C 8

```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
```

```
                                                                         L  8
**FILE**ID**DMDRIVER




            DDDDDDDD    MM      MM  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
            DDDDDDDD    MM      MM  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
            DD      DD  MMMM  MMMM  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
            DD      DD  MMMM  MMMM  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
            DD      DD  MM  MM  MM  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
            DD      DD  MM  MM  MM  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
            DD      DD  MM      MM  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
            DD      DD  MM      MM  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
            DD      DD  MM      MM  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
            DD      DD  MM      MM  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
            DD      DD  MM      MM  DD      DD  RR    RR        II        VV  VV    EE          RR    RR       ....
            DD      DD  MM      MM  DD      DD  RR    RR        II        VV  VV    EE          RR    RR       ....
            DDDDDDDD    MM      MM  DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR     ....
            DDDDDDDD    MM      MM  DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR     ....

            LL            IIIIII      SSSSSSSS
            LL            IIIIII      SSSSSSSS
            LL              II      SS
            LL              II      SS
            LL              II      SS
            LL              II      SS
            LL              II        SSSSSS
            LL              II        SSSSSS
            LL              II              SS
            LL              II              SS
            LL              II              SS
            LL              II              SS
            LLLLLLLLLL    IIIIII    SSSSSSSS
            LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  DMDRIVER - RK611-RK06/RK07 DISK DRIVER
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ; D. N. CUTLER 12-MAR-77
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-011 RAS0300         Ron Schaefer            27-Apr-1984
0000    ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    ;               devices will have the "node$" prefix.
0000    ;
0000    ;       V03-010 PRD0066         Paul R. DeStefano       24-Feb-1984
0000    ;               Modify DM$INT, RETREG, and DM_UNSOLNT to compensate for
0000    ;               the RK611 controller's failure to properly set/clear
0000    ;               volume valid bit in drive status register.
0000    ;
0000    ;       V03-009 WHM0001         Bill Matthews           22-Feb-1984
0000    ;               Fix a MOVL IDB$W_UNITS(R3),RO to be a MOVZWL IDB$W_UNITS(R3),RO
0000    ;               in routine GET_UNITS.
0000    ;
0000    ;       V03-008 PRD0045         Paul R. DeStefano       11-Jan-1984
0000    ;               Fix BBS instruction in DEVICE TIME OUT routine.
0000    ;
0000    ;       V03-007 PRD0032         Paul R. DeStefano       09-Sep-1983
0000    ;               Added EXE$LCLDSKVALID to function decision table.
0000    ;
0000    ;       V03-006 ROW0211         Ralph O. Weber          16-AUG-1983
0000    ;               Change device-dependent UCB definition base from UCB$W_BCR+2
0000    ;               to UCB$K_LCL_DISK_LENGTH.
0000    ;
0000    ;       V03-005 PRD0024         Paul R. DeStefano       06-May-1983
0000    ;               Modified RETREG routine to attempt to clear a drive
0000    ;               unsafe condidtion.
0000    ;
```

```
0000    ;
0000    ;         V03-004 PRD0019         Paul R. DeStefano        26-Apr-1983
0000    ;                 Modified FATALERR routine to return SS$_PARITY only for
0000    ;                 errors that possibly indicate bad media.  All other error
0000    ;                 conditions which formerly returned SS$_PARITY now return
0000    ;                 SS$_CNTLERR.
0000    ;
0000    ;         V03-003 PRD0016         Paul R. DeStefano        26-Apr-1983
0000    ;                 Modified ECC correction logic so that ECC is only applied
0000    ;                 when there is single bit ECC correctable error, or if there
0000    ;                 is a multiple bit ECC correctable error and the error cannot
0000    ;                 be corrected using retries.
0000    ;
0000    ;         V03-002 KDM0002         Kathleen D. Morse        28-Jun-1982
0000    ;                 Added $DCDEF, $DYNDEF, $PRDEF, and $SSDEF.
0000    ;
0000    ;         V03-001 KTA0100         Kerbey T. Altmann        07-Jun-1982
0000    ;                 Add code to set UCB$L_MEDIA_ID.
0000    ;
0000    ;
0000    ;**
```

```
0000    ; RK611-RK06/RK07 DISK DRIVER
0000    ;
0000    ; MACRO LIBRARY CALLS
0000    ;
0000
0000            $ADPDEF                                 ;DEFINE ADP OFFSETS
0000            $ACFDEF                                 ;DEFINE ACF OFFSETS
0000            $CRBDEF                                 ;DEFINE CRB OFFSETS
0000            $DCDEF                                  ;DEFINE DEVICE CLASSES
0000            $DEVDEF                                 ;DEFINE DEVICE CHARACTERISTICS BITS
0000            $DDBDEF                                 ;DEFINE DDB OFFSETS
0000            $DPTDEF                                 ;DEFINE DPT OFFSETS
0000            $DYNDEF                                 ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000            $EMBDEF                                 ;DEFINE EMB OFFSETS
0000            $IDBDEF                                 ;DEFINE IDB OFFSETS
0000            $IPLDEF                                 ;DEFINE USEFUL IPLs
0000            $IODEF                                  ;DEFINE I/O FUNCTION CODES
0000            $IRPDEF                                 ;DEFINE IRP OFFSETS
0000            $PRDEF                                  ;DEFINE PROCESSOR REGISTERS
0000            $SSDEF                                  ;DEFINE SYSTEM STATUS CODES
0000            $UCBDEF                                 ;DEFINE UCB OFFSETS
0000            $VECDEF                                 ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000
0000    ;
0000    ; LOCAL MACROS
0000    ;
0000    ; EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000    ;
0000
0000            .MACRO  EXFUNCH BDST,FCODE
0000                    .IF NB  FCODE
0000                    MOVZBL  #CD'FCODE,R3
0000                    .ENDC
0000                    BSBW    FEXH
0000                    .SIGNED_BYTE BDST-.-1
0000            .ENDM
0000
0000            .MACRO  EXFUNCL BDST,FCODE
0000                    .IF NB  FCODE
0000                    MOVZBL  #CD'FCODE,R3
0000                    .ENDC
0000                    BSBW    FEXL
0000                    .SIGNED_BYTE BDST-.-1
0000            .ENDM
0000
0000    ;
0000    ; GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000    ;
0000
0000            .MACRO  GENF FCODE
0000                    CD'FCODE=.-FTAB/2
0000                    .WORD   FCODE!RK_CS1_M_GO!RK_CS1_M_IE
0000            .ENDM
0000
0000    ;
0000    ; LOCAL SYMBOLS
0000    ;
```

```
0000    ; RK611-RK06/RK07 CONTROLLER REGISTER OFFSETS
0000    ;
0000
0000                    $DEFINI RK
0000
0000    $DEF            RK_CS1                  .BLKW   1       ;CONTROL STATUS REGISTER 1
0002                    _VIELD  RK_CS1,0,<-                     ; CONTROL STATUS REGISTER 1 FIELD DEFINITION
0002                            <GO,,M>,-                       ; GO BIT
0002                            <FCODE,4>,-                     ; FUNCTION CODE
0002                            <DPPE,,M>,-                     ; DATA PATH PURGE ERROR
0002                            <IE,,M>,-                       ; INTERRUPT ENABLE
0002                            <RDY,,M>,-                      ; CONTROLLER READY
0002                            <MEX,2>,-                       ; MEMORY EXTENSION BITS
0002                            <CDT,,M>,-                      ; CONTROLLER DRIVE TYPE
0002                            <CTO,,M>,-                      ; CONTROLLER TIME OUT
0002                            <CFMT,,M>,-                     ; CONTROLLER FORMAT ERROR
0002                            <SPAR,,M>,-                     ; SERIAL BUS PARITY ERROR
0002                            <DI,,M>,-                       ; DRIVE INTERRUPT
0002                            <CERR,,M>-                      ; CONTROLLER ERROR
0002                    >
0002    $DEF            RK_WC                   .BLKW   1       ;WORD COUNT REGISTER
0004    $DEF            RK_BA                   .BLKW   1       ;BUFFER ADDRESS REGISTER
0006    $DEF            RK_DA                   .BLKW   1       ;DESIRED SECTOR/TRACK ADDRESS REGISTER
0008                    _VIELD  RK_DA,0,<-                      ; DESIRED ADDRESS FIELD DEFINITIONS
0008                            <SA,5>,-                        ; DESIRED SECTOR ADDRESS
0008                            <,3>,-                          ; RESERVED BITS
0008                            <TA,3>-                         ; DESIRED TRACK ADDRESS
0008                    >
0008    $DEF            RK_CS2                  .BLKW   1       ;CONTROL STATUS REGISTER 2
000A                    _VIELD  RK_CS2,0,<-                     ; CONTROL STATUS REGISTER 2 FIELD DEFINITION
000A                            <DS,3>,-                        ; DRIVE SELECT
000A                            <RLS,,M>,-                      ; RELEASE DRIVE
000A                            <BAI,,M>,-                      ; BUFFER ADDRESS INCREMENT INHIBIT
000A                            <SCLR,,M>,-                     ; SUBSYSTEM CLEAR
000A                            <IR,,M>,-                       ; INPUT READY
000A                            <OR,,M>,-                       ; OUTPUT READY
000A                            <UFE,,M>,-                      ; UNIT FIELD ERROR
000A                            <MDS,,M>,-                      ; MULTIPLE DRIVE SELECT
000A                            <PGE,,M>,-                      ; PROGRAMMING ERROR
000A                            <NEM,,M>,-                      ; NONEXISTENT MEMORY
000A                            <NED,,M>,-                      ; NONEXISTENT DRIVE
000A                            <UPE,,M>,-                      ; UNIBUS PARITY ERROR
000A                            <WCE,,M>,-                      ; WRITE CHECK ERROR
000A                            <DLT,,M>-                       ; DATA LATE ERROR
000A                    >                                       ;
000A    $DEF            RK_DS                   .BLKW   1       ;DRIVE STATUS REGISTER
000C                    _VIELD  RK_DS,0,<-                      ; DRIVE STATUS REGISTER BIT DEFINTIONS
000C                            <DRA,,M>,-                      ; DRIVE AVAILABLE
000C                            <,1>,-                          ; RESERVED BIT
000C                            <OFST,,M>,-                     ; DRIVE OFFSET
000C                            <ACLO,,M>,-                     ; DRIVE AC LOW
000C                            <DCLO,,M>,-                     ; DRIVE DC LOW
000C                            <DROT,,M>,-                     ; DRIVE OFF TRACK
000C                            <VV,,M>,-                       ; VOLUME VALID
000C                            <DRDY,,M>,-                     ; DRIVE READY
000C                            <DDT,,M>,-                      ; DRIVE DRIVE TYPE
000C                            <,2>,-                          ; RESERVED BITS
```

```
000C                                <WRL,,M>,-                      ; DRIVE WRITE LOCKED
000C                                <,1>,-                          ; RESERVED BIT
000C                                <PIP,,M>,-                      ; POSITIONING IN PROGRESS
000C                                <DSC,,M>,-                      ; DRIVE STATUS CHANGE
000C                                <SVAL,,M>-                      ; DRIVE STATUS VALID
000C                       >                                        ;
000C       $DEF      RK_ER              .BLKW    1                  ;ERROR REGISTER
000E                 _VIELD   RK_ER,0,<-                            ; ERROR REGISTER BIT DEFINITIONS
000E                                <ILF,,M>,-                      ; ILLEGAL FUNCTION
000E                                <SKI,,M>,-                      ; SEEK INCOMPLETE
000E                                <NXF,,M>,-                      ; NONEXECUTABLE FUNCTION
000E                                <DRPAR,,M>,-                    ; DRIVE PARITY ERROR
000E                                <FMTE,,M>,-                     ; FORMAT ERROR
000E                                <DTYE,,M>,-                     ; DRIVE TYPE ERROR
000E                                <ECH,,M>,-                      ; ECC HARD ERROR
000E                                <BSE,,M>,-                      ; BAD SECTOR ERROR
000E                                <HVRC,,M>,-                     ; HEADER VRC ERROR
000E                                <COE,,M>,-                      ; CYLINDER OVERFLOW ERROR
000E                                <IDAE,,M>,-                     ; INVALID DISK ADDRESS ERROR
000E                                <WLE,,M>,-                      ; WRITE LOCK ERROR
000E                                <DTE,,M>,-                      ; DRIVE TIMING ERROR
000E                                <OPI,,M>,-                      ; OPERATION INCOMPLETE
000E                                <UNS,,M>,-                      ; DRIVE UNSAFE
000E                                <DCK,,M>-                       ; DATA CHECK ERROR
000E                 >                                              ;
000E       $DEF      RK_AS              .BLKW    1                  ;ATTENTION SUMMARY/OFFSET REGISTER
0010                 _VIELD   RK_AS,0,<-                            ; ATTENSION SUMMARY/OFFSET REGISTER FIELDS
0010                                <OF,7>,-                        ; DRIVE OFFSET
0010                                <,1>,-                          ; RESERVED BIT
0010                                <ATTN,8,M>-                     ; DRIVE ATTENTION SUMARY
0010                 >                                              ;
0010       $DEF      RK_DC              .BLKW    1                  ;DESIRED CYLINDER ADDRESS
0012       $DEF      RK_SPR             .BLKW    1                  ;UNUSED REGISTER
0014       $DEF      RK_DB              .BLKW    1                  ;DATA BUFFER REGISTER
0016       $DEF      RK_MR1             .BLKW    1                  ;MAINTENANCE REGISTER 1
0018                 _VIELD   RK_MR1,0,<<MS,3>>                     ; MAINTENANCE REGISTER 1 FIELD DEFINITION
0018       $DEF      RK_EC1             .BLKW    1                  ;ECC POSITION REGISTER
001A                 _VIELD   RK_EC1,0,<<EPS,13>>                   ; ECC POSITION FIELD
001A       $DEF      RK_EC2             .BLKW    1                  ;ECC PATTERN REGISTER
001C                 _VIELD   RK_EC2,0,<<EPT,11>>                   ; ECC PATTERN FIELD
001C       $DEF      RK_MR2             .BLKW    1                  ;MAINTENANCE REGISTER 2
001E       $DEF      RK_MR3             .BLKW    1                  ;MAINTENANCE REGISTER 3
0020
0020                 $DEFEND RK
0000
0000       ;
0000       ; SOFTWARE STATUS IN UPPER BYTE OF OFFSET WORD
0000       ;
0000
0000                 _VIELD   DM,0,<-                               ;SOFTWARE STATUS BIT DEFINTIONS
0000                                <ECI,,M>,-                      ; ECC INHIBIT
0000                                <DCK,,M>-                       ; DATACHECK IN PROGRESS
0000                                <ECC_DEFER,,M>, -               ; Flag to indicate that ECC correction
0000                 >                                              ;  has been deferred until offset
0000                                                                ;  retries are exhausted.
0000
0000       ;
```

```
                0000  ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK  OFFSETS
                0000  ;
                (000
                0000          $DEFINI UCB
                0000
      000000CC  0000  .=UCB$K_LCL_DISK_LENGTH                   ; Establish device-dependent base
                00CC
                00CC  $DEF    UCB$W_DM_DTYP   .BLKW   1         ;DRIVE TYPE MASK
                00CE  $DEF    UCB$W_DM_CS1    .BLKW   1         ;CONTROL STATUS REGISTER 1
                00D0  $DEF    UCB$W_DM_WC     .BLKW   1         ;WORK COUNT REGISTER
                00D2  $DEF    UCB$W_DM_BA     .BLKW   1         ;BUFFER ADDRESS REGISTER
                00D4  $DEF    UCB$W_DM_DA     .BLKW   1         ;DISK ADDRESS REGISTER
                00D6  $DEF    UCB$W_DM_CS2    .BLKW   1         ;CONTROL STATUS REGISTER 2
                00D8  $DEF    UCB$W_DM_DS     .BLKW   1         ;DRIVE STATUS REGISTER
                00DA  $DEF    UCB$W_DM_ER     .BLKW   1         ;ERROR REGISTER
                00DC  $DEF    UCB$W_DM_AS     .BLKW   1         ;ATTENTION SUMMARY REGISTER
                00DE  $DEF    UCB$W_DM_DC     .BLKW   1         ;DESIRED CYLINDER REGISTER
                00E0  $DEF    UCB$W_DM_MR1    .BLKW   1         ;MAINTENANCE REGISTER 1
                00E2  $DEF    UCB$W_DM_MR2    .BLKW   1         ;MAINTENANCE REGISTER 2
                00E4  $DEF    UCB$W_DM_MR3    .BLKW   1         ;MAINTENANCE REGISTER 3
                00E6  $DEF    UCB$W_DM_DPN    .BLKW   1         ;DATAPATH NUMBER
                00E8  $DEF    UCB$L_DM_DPR    .BLKL   1         ;DATAPATH REGISTER
                00EC  $DEF    UCB$L_DM_FMPR   .BLKL   1         ;FINAL MAP REGISTER
                00F0  $DEF    UCB$L_DM_PMPR   .BLKL   1         ;PREVIOUS MAP REGISTER
                00F4  $DEF    UCB$W_DM_DB     .BLKW   3         ;DATA BUFFER REGISTER
                00FA  $DEF    UCB$B_DM_IND    .BLKB   1         ;SOFTWARE INDICATORS
                00FB          _VIELD  DM_IND,0,<-               ;INDICATOR BIT DEFINITIONS
                00FB                  <OF,,M>-                  ; OFFSET FLAG
                00FB          >                                 ;
                00FB  $DEF    UCB$L_DM_FRS    .BLKL   1         ;FINAL REQUEST STATUS
      00000100  00FF                          .BLKB   1         ;SPARE USED BYTE
                0100
      00000100  0100  UCB$K_DM_LENGTH=.
                0100
                0100          $DEFEND UCB
                0000
                0000  ;
                0000  ; HARDWARE FUNCTION CODES
                0000  ;
                0000
      00000000  0000  F_NOP=0*2                                 ;NO OPERATION (SELECT DRIVE)
      00000006  0000  F_UNLOAD=3*2                              ;UNLOAD DRIVE
      0000000E  0000  F_SEEK=7*2                                ;SEEK CYLINDER
      0000000A  0000  F_RECAL=5*2                               ;RECALIBRATE
      00000004  0000  F_DRVCLR=2*2                              ;DRIVE CLEAR
      00000000  0000  F_RELEASE=0*2                             ;RELEASE DRIVE
      0000000C  0000  F_OFFSET=6*2                              ;OFFSET HEADS
      0000000C  0000  F_RETCENTER=6*2                           ;RETURN TO CENTERLINE
      00000002  0000  F_PACKACK=1*2                             ;PACK ACKNOWLEDGE
      00000008  0000  F_STARTSPNDL=4*2                          ;START SPINDLE
      00000018  0000  F_WRITECHECK=12*2                         ;WRITE CHECK DATA
      00000012  0000  F_WRITEDATA=9*2                           ;WRITE DATA
      00000016  0000  F_WRITEHEAD=11*2                          ;WRITE HEADER AND DATA
      00000010  0000  F_READDATA=8*2                            ;READ DATA
      00000014  0000  F_READHEAD=10*2                           ;READ HEADER AND DATA
      00000000  0000  F_AVAILABLE=F_NOP                         ;DRIVE AVAILABLE (a NOP)
                0000
```

```
0000  ;
0000  ; LOCAL DATA
0000  ;
0000  ; DRIVER PROLOGUE TABLE
0000  ;
0000
0000          DPTAB   -                           ;DEFINE DRIVER PROLOGUE TABLE
0000                  END=DM_END,-                ;END OF DRIVER
0000                  ADAPTER=UBA,-               ;ADAPTER TYPE
0000                  FLAGS=DPT$M_SVP,-           ;SYSTEM PAGE TABLE ENTRY REQUIRED
0000                  UCBSIZE=UCB$K_DM_LENGTH,-          ;UCB SIZE
0000                  DEFUNITS=8,-                ;Default number of AUTOCONFIGURE units
0000                  DELIVER=DM$DELIVER,-        ;AUTOCONFIGURE units delivery routine
0000                  NAME=DMDRIVER               ;DRIVER NAME
0038          DPT_STORE INIT                      ;CONTROL BLOCK INIT VALUES
0038          DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\> ;DEFAULT ACP NAME
003F          DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_CART ;ACP CLASS
0043          DPT_STORE UCB,UCB$B_FIPL,B,8        ;FORK IPL
0047          DPT_STORE UCB,UCB$L_DEVCHAR,L,-     ;DEVICE CHARACTERISTICS
0047                  <DEV$M_FOD-                 ; FILES ORIENTED
0047                  !DEV$M_DIR-                 ; DIRECTORY STRUCTURED
0047                  !DEV$M_AVL-                 ; AVAILABLE
0047                  !DEV$M_ELG-                 ; ERROR LOGGING ENABLED
0047                  !DEV$M_SHR-                 ; SHAREABLE
0047                  !DEV$M_IDV-                 ; INPUT DEVICE
0047                  !DEV$M_ODV-                 ; OUTPUT DEVICE
0047                  !DEV$M_RND>                 ; RANDOM ACCESS
004E          DPT_STORE UCB,OCB$L_DEVCHAR2,L,-;   DEVICE CHARACTERISTICS
004E                  <DEV$M_NNM>                 ; PREFIX NAME WITH "node$"
0055          DPT_STORE UCB,OCB$B_DEVCLASS,B,DC$_DISK ;DEVICE CLASS
0059          DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
005E          DPT_STORE UCB,UCB$B_SECTORS,B,22 ;NUMBER OF SECTORS PER TRACK
0062          DPT_STORE UCB,UCB$B_TRACKS,B,3   ;NUMBER OF TRACKS PER CYLINDER
0066          DPT_STORE UCB,UCB$B_DIPL,B,21       ;DEVICE IPL
006A          DPT_STORE UCB,UCB$B_ERTCNT,B,8      ;ERROR RETRY COUNT
006E          DPT_STORE UCB,UCB$B_ERTMAX,B,8      ;MAX ERROR RETRY COUNT
0072          DPT_STORE REINIT                    ;CONTROL BLOCK RE-INIT VALUES
0072          DPT_STORE CRB,CRB$L_INTD+4,D,DM$INT ;INTERRUPT SERVICE ROUTINE ADDRESS
0077          DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,D,DM_RK611_INIT ;CONTROLLER INIT
007C          DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,DM_RKOX_INIT ;UNIT INIT
0081          DPT_STORE DDB,DDB$L_DDT,D,DM$DDT ;DDT ADDRESS
0086          DPT_STORE END                       ;
0000
0000  ;
0000  ; DRIVER DISPATCH TABLE
0000  ;
0000
0000          DDTAB   DM,-                        ;DRIVER DISPATCH TABLE
0000                  DM_STARTIO,-                ;START I/O OPERATION
0000                  DM_UNSOLNT,-                ;UNSOLICITED INTERRUPT
0000                  DM_FUNCTABLE,-              ;FUNCTION DECISION TABLE
0000                  0,-                         ;CANCEL I/O ENTRY POINT
0000                  DM_REGDUMP,-                ;REGISTER DUMP ROUTINE
0000                  <<2RK_MR3+2-4+8>*2>+<<3+5+1>*4>>,- ;SIZE OF DIAGNOSTIC BUFFER
0000                  <<<RK_MR3+2-4+8>*2>+<1*4>+<EMB$L_DV_REGSAV>> ;SIZE OF ERROR BUFFER
0038
0038  ;
```

```
                    0038   365 ; HARDWARE I/O FUNCTION CODE TABLE
                    0038   366 ;
                    0038   367
                    0038   368 FTAB:                                          ;
                    0038   369         GENF    F_NOP                          ;NO OPERATION
                    003A   370         GENF    F_UNLOAD                       ;UNLOAD VOLUME
                    003C   371         GENF    F_SEEK                         ;SEEK CYLINDER
                    003E   372         GENF    F_RECAL                        ;RECALIBRATE
                    0040   373         GENF    F_DRVCLR                       ;DRIVE CLEAR
                    0042   374         GENF    F_RELEASE                      ;RELEASE PORT
                    0044   375         GENF    F_OFFSET                       ;OFFSET HEADS
                    0046   376         GENF    F_RETCENTER                    ;RETURN HEADS TO CENTERLINE
                    0048   377         GENF    F_PACKACK                      ;PACK ACKNOWLEDGE
                    004A   378         GENF    F_STARTSPNDL                   ;START SPINDLE
                    004C   379         GENF    F_WRITECHECK                   ;WRITE CHECK
                    004E   380         GENF    F_WRITEDATA                    ;WRITE DATA
                    0050   381         GENF    F_READDATA                     ;READ DATA
                    0052   382         GENF    F_WRITEHEAD                    ;WRITE HEADERS
                    0054   383         GENF    F_READHEAD                     ;READ HEADER
                    0056   384         GENF    F_AVAILABLE                    ;DRIVE AVAILABLE
                    0058   385
                    0058   386 ;
                    0058   387 ; OFFSET TABLE FOR RK611-RK06/RK07
                    0058   388 ;
                    0058   389
                    0058   390 OFFTAB:                                        ;
              00    0058   391         .BYTE   0                              ;RETURN TO CENTERLINE
              10    0059   392         .BYTE   ^X10                           ;+400
              90    005A   393         .BYTE   ^X90                           ;-400
              20    005B   394         .BYTE   ^X20                           ;+800
              A0    005C   395         .BYTE   ^XA0                           ;-800
              30    005D   396         .BYTE   ^X30                           ;+1200
              B0    005E   397         .BYTE   ^XB0                           ;-1200
              00    005F   398         .BYTE   0                              ;RETURN TO CENTERLINE
        00000008    0060   399 OFFSIZ=.-OFFTAB                                ;SIZE OF OFFSET TABLE
```

```
0060            .SBTTL  RK611-RK06/RK07 FUNCTION DECISION TABLE
0060    ;+
0060    ; RK611-RK06/RK07 FUNCTION DECISION TABLE
0060    ;-
0060
0060    DM_FUNCTABLE:                           ;FUNCTION DECISION TABLE
0060            FUNCTAB ,-                      ;LEGAL FUNCTIONS
0060                    <NOP,-                  ;NO OPERATION
0060                    UNLOAD,-                ;UNLOAD VOLUME
0060                    SEEK,-                  ;SEEK CYLINDER
0060                    RECAL,-                 ;RECALIBRATE
0060                    DRVCLR,-                ;DRIVE CLEAR
0060                    RELEASE,-               ;RELEASE PORT
0060                    OFFSET,-                ;OFFSET HEADS
0060                    RETCENTER,-             ;RETURN HEADS TO CENTERLINE
0060                    PACKACK,-               ;PACK ACKNOWLEDGE
0060                    AVAILABLE,-             ;DRIVE AVAILABLE
0060                    STARTSPNDL,-            ;START SPINDLE
0060                    SENSECHAR,-             ;SENSE CHARACTERISTICS
0060                    SETCHAR,-               ;SET CHARACTERISITCS
0060                    SENSEMODE,-             ;SENSE MODE
0060                    SETMODE,-               ;SET MODE
0060                    WRITECHECK,-            ;WRITE CHECK
0060                    WRITEHEAD,-             ;WRITE HEADERS
0060                    READHEAD,-              ;READ HEADER
0060                    READLBLK,-              ;READ LOGICAL BLOCK
0060                    WRITELBLK,-             ;WRITE LOGICAL BLOCK
0060                    READPBLK,-              ;READ PHYSICAL BLOCK
0060                    WRITEPBLK,-             ;WRITE PHYSICAL BLOCK
0060                    READVBLK,-              ;READ VIRTUAL BLOCK
0060                    WRITEVBLK,-             ;WRITE VIRTUAL BLOCK
0060                    ACCESS,-                ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
0060                    ACPCONTROL,-            ;ACP CONTROL FUNCTION
0060                    CREATE,-                ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
0060                    DEACCESS,-              ;DEACCESS FILE
0060                    DELETE,-                ;DELETE FILE AND/OR DIRECTORY ENTRY
0060                    MODIFY,-                ;MODIFY FILE ATTRIBUTES
0060                    MOUNT>                  ;MOUNT VOLUME
0068            FUNCTAB ,-                      ;BUFFERED I/O FUNCTIONS
0068                    <NOP,-                  ;NO OPERATION
0068                    UNLOAD,-                ;UNLOAD VOLUME
0068                    SEEK,-                  ;SEEK CYLINDER
0068                    RECAL,-                 ;RECALIBRATE
0068                    DRVCLR,-                ;DRIVE CLEAR
0068                    RELEASE,-               ;RELEASE PORT
0068                    OFFSET,-                ;OFFSET HEADS
0068                    RETCENTER,-             ;RETURN HEADS TO CENTERLINE
0068                    PACKACK,-               ;PACK ACKNOWLEDGE
0068                    AVAILABLE,-             ;DRIVE AVAILABLE
0068                    STARTSPNDL,-            ;START SPINDLE
0068                    SENSECHAR,-             ;SENSE CHARACTERISTICS
0068                    SETCHAR,-               ;SET CHARACTERISITCS
0068                    SENSEMODE,-             ;SENSE MODE
0068                    SETMODE,-               ;SET MODE
0068                    ACCESS,-                ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
0068                    ACPCONTROL,-            ;ACP CONTROL FUNCTION
0068                    CREATE,-                ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
```

```
0068                 DEACCESS,-                ;DEACCESS FILE
0068                 DELETE,-                  ;DELETE FILE AND/OR DIRECTORY ENTRY
0068                 MODIFY,-                  ;MODIFY FILE ATTRIBUTES
0068                 MOUNT>                    ;MOUNT VOLUME
0070        FUNCTAB DM_BYTECNT,-               ;EVEN BYTE COUNT REQUIRED FUNCTIONS
0070                 <READHEAD,-               ;READ HEADER
0070                 READLBLK,-                ;READ LOGICAL BLOCK
0070                 READPBLK,-                ;READ PHYSICAL BLOCK
0070                 READVBLK,-                ;READ VIRTUAL BLOCK
0070                 WRITECHECK,-              ;WRITE CHECK
0070                 WRITEHEAD,-               ;WRITE HEADERS
0070                 WRITELBLK,-               ;WRITE LOGICAL BLOCK
0070                 WRITEPBLK,-               ;WRITE PHYSICAL BLOCK
0070                 WRITEVBLK>                ;WRITE VIRTUAL BLOCK
007C        FUNCTAB +ACP$READBLK,-             ;READ FUNCTIONS
007C                 <READHEAD,-               ;READ HEADER
007C                 READLBLK,-                ;READ LOGICAL BLOCK
007C                 READPBLK,-                ;READ PHYSICAL BLOCK
007C                 READVBLK>                 ;READ VIRTUAL BLOCK
0088        FUNCTAB +ACP$WRITEBLK,-            ;WRITE FUNCTIONS
0088                 <WRITECHECK,-             ;WRITE CHECK
0088                 WRITEHEAD,-               ;WRITE HEADERS
0088                 WRITELBLK,-               ;WRITE LOGICAL BLOCK
0088                 WRITEPBLK,-               ;WRITE PHYSICAL BLOCK
0088                 WRITEVBLK>                ;WRITE VIRTUAL BLOCK
0094        FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00A0        FUNCTAB +ACP$DEACCESS,<DEACCESS>  ;DEACCESS FILE
00AC        FUNCTAB +ACP$MODIFY,-              ;
00AC                 <ACPCONTROL,-             ;ACP CONTROL FUNCTION
00AC                 DELETE,-                  ;DELETE FILE OR DIRECTORY ENTRY
00AC                 MODIFY>                   ;MODIFY FILE ATTRIBUTES
00B8        FUNCTAB +ACP$MOUNT,<MOUNT>         ;MOUNT VOLUME
00C4        FUNCTAB +EXE$LCLDSKVALID,-         ;LOCAL DISK VALID FUNCTIONS
00C4                 <UNLOAD,-                 ;UNLOAD VOLUME
00C4                 AVAILABLE,-               ;UNIT AVAILABLE
00C4                 PACKACK>                  ;PACK ACKNOWLEDGE
00D0        FUNCTAB +EXE$ZEROPARM,-            ;ZERO PARAMETER FUNCTIONS
00D0                 <NOP,-                    ;NO OPERATION
00D0                 UNLOAD,-                  ;UNLOAD VOLUME
00D0                 RECAL,-                   ;RECALIBRATE
00D0                 DRVCLR,-                  ;DRIVE CLEAR
00D0                 RELEASE,-                 ;RELEASE PORT
00D0                 RETCENTER,-               ;RETURN HEADS TO CENTERLINE
00D0                 STARTSPNDL,-              ;START SPINDLE
00D0                 PACKACK,-                 ;PACK ACKNOWLEDGE
00D0                 AVAILABLE>                ;DRIVE AVAILABLE
00DC        FUNCTAB +EXE$ONEPARM,-             ;ONE PARAMETER FUNCTIONS
00DC                 <SEEK,-                   ;SEEK CYLINDER
00DC                 OFFSET>                   ;OFFSET HEADS
00E8        FUNCTAB +EXE$SENSEMODE,-           ;
00E8                 <SENSECHAR,-              ;SENSE CHARACTERISTICS
00E8                 SENSEMODE>                ;SENSE MODE
00F4        FUNCTAB +EXE$SETCHAR,-             ;
00F4                 <SETCHAR,-                ;SET CHARACTERISTICS
00F4                 SETMODE>                  ;SET MODE
```

```
                             0100            .SBTTL  TEST EVEN BYTE COUNT
                             0100      ;+
                             0100      ; DM_BYTECNT - TEST EVEN BYTE COUNT
                             0100      ;
                             0100      ; THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER TO CHECK
                             0100      ; THAT THE NUMBER OF BYTES TO BE TRANSFERED IS EVEN AS THE RK611 HAS A WORD
                             0100      ; COUNT RATHER THAN A BYTE COUNT REGISTER.
                             0100      ;
                             0100      ; INPUTS:
                             0100      ;
                             0100      ;       R0 = SCRATCH.
                             0100      ;       R1 = SCRATCH.
                             0100      ;       R2 = SCRATCH.
                             0100      ;       R3 = ADDRESS OF I/O REQUEST PACKET.
                             0100      ;       R4 = CURRENT PROCESS PCB ADDRESS.
                             0100      ;       R5 = ASSIGNED DEVICE UCB ADDRESS.
                             0100      ;       R6 = ADDRESS OF CCB.
                             0100      ;       R7 = I/O FUNCTION CODE.
                             0100      ;       R8 = FUNCTION DECISION TABLE DISPATCH ADDRESS.
                             0100      ;       R9 = SCRATCH.
                             0100      ;       R10 = SCRATCH.
                             0100      ;       R11 = SCRATCH.
                             0100      ;       AP = ADDRESS OF FIRST FUNCTION DEPENDENT PARAMETER.
                             0100      ;
                             0100      ; OUTPUTS:
                             0100      ;
                             0100      ;       THE BUFFER BYTE COUNT IS CHECKED FOR BEING EQUAL TO 0 MOD 2. IF THE CHECK
                             0100      ;       FAILS, THEN THE I/O OPERATION IS TERMINATED WITH AN ERROR. ELSE A RETURN
                             0100      ;       TO THE FUNCTION DECISION TABLE DISPATCHER IS EXECUTED.
                             0100      ;-
                             0100
                             0100     DM_BYTECNT:                                   ;
    01 04 AC      00    E0   0100            BBS     #0,4(AP),10$                   ;IF SET, ODD BYTE COUNT
                        05   0105            RSB                                    ;
    50    034C 8F       3C   0106     10$:   MOVZWL  #SS$_IVBUFLEN,R0               ;SET ODD BYTE COUNT STATUS
       00000000'GF      17   010B            JMP     G^EXE$ABORTIO                  ;
```

```
                              0111         .SBTTL  START I/O OPERATION
                              0111  ;+
                              0111  ; STARTIO - START I/O OPERATION ON DEVICE UNIT
                              0111  ;
                              0111  ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
                              0111  ;
                              0111  ; INPUTS:
                              0111  ;
                              0111  ;       R3 = ADDRESS OF I/O PACKET.
                              0111  ;       R5 = UCB ADDRESS OF DEVICE UNIT.
                              0111  ;
                              0111  ; OUTPUTS:
                              0111  ;
                              0111  ;       FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
                              0111  ;       RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
                              0111  ;       THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
                              0111  ;-
                              0111
                              0111  DM_STARTIO:                               ;START I/O OPERATION
0080 C5    0081 C5    90      0111          MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
  009A C5    20 A3    B0      0118          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
       50    38 A3    D0      011E          MOVL    IRP$L_MEDIA(R3),R0         ;GET PARAMETER LONGWORD
           00FA C5    94      0122          CLRB    UCB$B_DM_IND(R5)           ;CLEAR SOFTWARE INDICATOR BYTE
                04    8A      0126          BICB    #DM_M_ECC_DEFER,-          ; Clear flag used to signal ECC
           00C9 C5            0128                  UCB$W_OFFSET+1(R5)         ;  correction deferred.
                              012B
                              012B  ;
                              012B  ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                              012B  ;
                              012B
           06    00    EF     012B  10$:    EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODE
       51    20 A3            012E                  IRP$W_FUNC(R3),R1          ;
           51    02    91     0131          CMPB    #IO$_SEEK,R1               ;SEEK FUNCTION?
                 23    13     0134          BEQL    20$                        ;IF EQL YES
           51    07    91     0136          CMPB    #IO$_RETCENTER,R1          ;RETURN HEADS TO CENTERLINE?
                 25    13     0139          BEQL    30$                        ;IF EQL YES
           51    06    91     013B          CMPB    #IO$_OFFSET,R1             ;OFFSET FUNCTION?
                 22    13     013E          BEQL    40$                        ;IF EQL YES
      00BC C5    50    D0     0140          MOVL    R0,UCB$W_DA(R5)            ;STORE PARAMETER LONGWORD
           51    19    91     0145          CMPB    #IO$_STARTSPNDL,R1         ;Check for IO$_STARTSPNDL
                 0A    13     0148          BEQL    16$                        ;and
           51    11    91     014A          CMPB    #IO$_AVAILABLE,R1          ;IO$_AVAILABLE, the two function
                 1D    12     014D          BNEQ    50$                        ;codes which have different internal
                              014F                                             ;values.
           51    0F    90     014F          MOVB    #CDF_AVAILABLE,R1          ;Map IO$_AVAILABLE to CDF_AVAILABLE.
                 18    11     0152          BRB     50$
           51    09    90     0154  16$:    MOVB    #CDF_STARTSPNDL,R1         ;Map IO$_STARTSPNDL to CDF_STARTSPNDL.
                 13    11     0157          BRB     50$
                              0159
                              0159  ;
                              0159  ; SEEK FUNCTION - SET CYLINDER ADDRESS
                              0159  ;
                              0159
      00BE C5    50    B0     0159  20$:    MOVW    R0,UCB$W_DC(R5)            ;SET CYLINDER ADDRESS
                 0C    11     015E          BRB     50$                        ;
                              0160
                              0160  ;
```

```
                        0160    ; RETURN HEADS TO CENTERLINE FUNCTION - CLEAR OFFSET VALUE
                        0160    ;
                        0160
               50   D4  0160    30$:    CLRL    R0                              ;CLEAR OFFSET VALUE
                        0162
                        0162    ;
                        0162    ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
                        0162    ;
                        0162
      00C8 C5  50   90  0162    40$:    MOVB    R0,UCB$W_OFFSET(R5)     ;SET OFFSET VALUE
      00FA C5  01   88  0167            BISB    #DM_IND_M_OF,UCB$B_DM_IND(R5)   ;SET OFFSET FLAG
                        016C
                        016C    ;
                        016C    ; FINISH PREPROCESSING
                        016C    ;
                        016C
      0092 C5  51   90  016C    50$:    MOVB    R1,UCB$B_FEX(R5)        ;SAVE FUNCTION DISPATCH INDEX
        68 A5  03   AA  0171            BICW    #UCB$M_ECC!-            ;CLEAR ECC CORRECTION MADE AND,
                        0175            UCB$M_DIAGBUF,UCB$W_DEVSTS(R5) ;DIAGNOSTIC BUFFER PRESENT
     04 2A A3  07   E1  0175            BBC     #IRP$V_DIAGBUF,IRP$W_STS(R3),FDISPATCH ;IF CLR, NO BUFFER
        68 A5  02   A8  017A            BISW    #UCB$M_DIAGBUF,UCB$W_DEVSTS(R5) ;SET DIAGNOSTIC BUFFER PRESENT
                        017E
                        017E    ;
                        017E    ; CENTRAL FUNCTION DISPATCH
                        017E    ;
                        017E
                        017E    FDISPATCH:                              ;FUNCTION DISPATCH
        53    58 A5 D0  017E            MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF I/O PACKET
     0D 2A A3  08   E0  0182            BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
     08 64 A5  08   E0  0187            BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
     50   0254 8F   3C  018C            MOVZWL  #SS$_VOLINV,R0          ;SET VOLUME INVALID STATUS
             0590   31  0191            BRW     RESETXFR                ;
                        0194
                        0194    ;
                        0194    ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
                        0194    ;
                        0194
         00C9 C5    94  0194    10$:    CLRB    UCB$W_OFFSET+1(R5)      ;CLEAR ECC INHIBIT AND DATACHECK IN PROGRESS
      00CB C5  01   90  0198            MOVB    #1,UCB$B_OFFRTC(R5)     ;SET INITIAL OFFSET RETRY COUNT
         00CA C5    94  019D            CLRB    UCB$B_OFFNDX(R5)        ;CLEAR INITIAL OFFSET TABLE INDEX
     53  0092 C5    9A  01A1            MOVZBL  UCB$B_FEX(R5),R3        ;GET FUNCTION DISPATCH INDEX
                        01A6            CASE    R3,<-                   ;DISPATCH TO FUNCTION HANDLING ROUTINE
                        01A6                    NOP,-                   ;NO OPERATION
                        01A6                    UNLOAD,-                ;UNLOAD DRIVE
                        01A6                    SEEK,-                  ;SEEK CYLINDER
                        01A6                    RECAL,-                 ;RECALIBRATE
                        01A6                    DRVCLR,-                ;DRIVE CLEAR
                        01A6                    RELEASE,-               ;RELEASE PORT
                        01A6                    OFFSET,-                ;OFFSET HEADS
                        01A6                    RETCENTER,-             ;RETURN TO CENTERLINE
                        01A6                    PACKACK,-               ;PACK ACKNOWLEDGE
                        01A6                    STARTSPNDL,-            ;START SPINDLE
                        01A6                    WRITECHECK,-            ;WRITE CHECK DATA
                        01A6                    WRITEDATA,-             ;WRITE DATA
                        01A6                    READDATA,-              ;READ DATA
                        01A6                    WRITEHEAD,-             ;WRITE HEADER
                        01A6                    READHEAD>               ;READ HEADER
                        01C8
                        01C8   ;
                        01C8   ; AVAILABLE -- Mark volume not valid
                        01C8   ;
                        01C8
                        01C8   AVAILABLE:                                      ;DRIVE AVAILABLE
   64 A5   0800 8F  AA  01C8           BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Mark volume "invalid."
                6E  11  01CE           BRB     NORMAL                          ;Complete I/O processing.
                        01D0
                        01D0   ;
                        01D0   ; UNLOAD -- Mark volume not valid
                        01D0   ;
                        01D0
                        01D0   UNLOAD:                                         ;UNLOAD DRIVE
   64 A5   0800 8F  AA  01D0           BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Mark volume "invalid."
                06  11  01D6           BRB     EXEC_FUNCTION                   ;Do hardware I/O operation.
                        01D8
                        01D8   ;
                        01D8   ; PACK ACKNOWLEDGE -- Mark volume valid
                        01D8   ;
                        01D8
                        01D8   PACKACK:                                        ;PACK ACKNOWLEDGE
   64 A5   0800 8F  A8  01D8           BISW    #UCB$M_VALID, UCB$W_STS(R5) ;Mark volume "valid."
                        01DE   ;       BRB     EXEC_FUNCTION                   ;Do hardware I/O operation.
                        01DE
                        01DE   ;
                        01DE   ; NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
                        01DE   ; RETURN TO CENTER LINE, AND START SPINDLE
                        01DE   ;
                        01DE
                        01DE   NOP:                                            ;NO OPERATION
                        01DE   SEEK:                                           ;SEEK CYLINDER
                        01DE   RECAL:                                          ;RECALIBRATE
                        01DE   DRVCLR:                                         ;DRIVE CLEAR
                        01DE   RELEASE:                                        ;RELEASE PORT
                        01DE   OFFSET:                                         ;OFFSET READ HEADS
                        01DE   RETCENTER:                                      ;RETURN TO CENTERLINE
                        01DE   STARTSPNDL:                                     ;START SPINDLE
                        01DE   EXEC_FUNCTION:
                        01DE           EXFUNCH 10$                             ;EXECUTE HOUSEKEEPING FUNCTION
                5A  11  01E2           BRB     NORMAL                          ;
                        01E4   10$:
              0080  31  01E4           BRW     RETRY                           ; Use BRW since EXFUNCH only allows
                        01E7                                                   ;  for byte offset.
                        01E7
                        01E7   ;
                        01E7   ; WRITE CHECK DATA, WRITE HEADERS, AND READ HEAD
                        01E7   ;
                        01E7
                        01E7   WRITECHECK:                                     ;WRITE CHECK DATA
                        01E7   WRITEHEAD:                                      ;WRITE HEADERS
                        01E7   READHEAD:                                       ;READ HEADER
 009A C5   4000 8F  AA  01E7           BICW    #IO$M_DATACHECK,UCB$W_FUNC(R5) ;CLEAR DATA CHECK REQUEST
                        01EE
                        01EE   ;
                        01EE   ; WRITE DATA, WRITE CHECK DATA, WRITE HEADERS, AND READ HEADER
                        01EE   ;
```

```
                            01EE
                            01EE  WRITEDATA:                                      ;WRITE DATA
      00C9 C5    01    88   01EE          BISB    #DM_M_ECI,UCB$W_OFFSET+1(R5) ;SET ECC INHIBIT
                            01F3
                            01F3  ;
                            01F3  ; READ DATA, WRITE DATA, WRITE CHECK DATA, WRITE HEADERS, AND READ HEADER
                            01F3  ;
                            01F3
                            01F3  READDATA:                                       ;READ DATA
   07 009A C5    0C    E0   01F3          BBS     #IO$V_INHSEEK,UCB$W_FUNC(R5),TRANSFR ;IF SET, NO EXPLICIT SEEK
                            01F9          EXFUNCH RETRY,F_SEEK                    ;SEEK DESIRED CYLINDER
                            0200
                            0200  ;
                            0200  ; DATA TRANSFER
                            0200  ;
                            0200
                            0200  TRANSFR:                                        ;DATA TRANSFER REQUEST CHANNEL
     53   0092 C5      9A   0200          MOVZBL  UCB$B_FEX(R5),R3                ;GET FUNCTION DISPATCH INDEX
                            0205          EXFUNCL TRANXT                          ;EXECUTE TRANSFER FUNCTION
                            0209
                            0209  ;
                            0209  ; DATA CHECK
                            0209  ;
                            0209
                            0209  DATACHECK:                                      ;DATA CHECK
   2F 009A C5    0E    E1   0209          BBC     #IO$V_DATACHECK,UCB$W_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
     50   0639 8F      3C   020F          MOVZWL  #SS$_WASECC,R0                  ;ASSUME ECC CORRECTION WAS MADE
   27 009A C5    00    E0   0214          BBS     #UCB$V_ECC,UCB$W_FUNC(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
      00C9 C5    03    90   021A          MOVB    #DM_M_DCK!-                     ;SET DATA CHECK IN PROGRESS
                            021F                  DM_M_ECI,UCB$W_OFFSET+1(R5) ;AND INHIBIT ECC CORRECTION
      00CB C5    01    90   021F          MOVB    #1,UCB$B_OFFRTC(R5)             ;SET INITIAL OFFSET RETRY COUNT
           00CA C5     94   0224          CLRB    UCB$B_OFFNDX(R5)                ;CLEAR INITIAL OFFSET TABLE INDEX
        52    58 A5    D0   0228          MOVL    UCB$L_IRP(R5),R2                ;GET ADDRESS OF IRP
     78 A5    2C A2    7D   022C          MOVQ    IRP$L_SVAPTE(R2),UCB$L_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
   00BC C5    38 A2    D0   0231          MOVL    IRP$L_MEDIA(R2),UCB$W_DA(R5) ;
                            0237
                            0237  ;
                            0237  ; DATA CHECK RETRY
                            0237  ;
                            0237
                            0237  CHECKRETRY:                                     ;DATA CHECK RETRY
                            0237          EXFUNCL TRANXT,F_WRITECHECK             ;EXECUTE WRITECHECK FUNCTION
                            023E
                            023E  ;
                            023E  ; SUCCESSFUL OPERATION COMPLETION
                            023E  ;
                            023E
                            023E  NORMAL:                                         ;
          50    01     3C   023E          MOVZWL  S^#SS$_NORMAL,R0                ;SET NORMAL COMPLETION STATUS
                            0241  CHECKXT:                                        ;
             017D      31   0241          BRW     FUNCXT                          ;
                            0244
                            0244  ;
                            0244  ; TRANSFER ENDED WITH A RETRIABLE ERROR
                            0244  ;
                            0244
                            0244  TRANXT:                                         ;TRANSFER EXIT
```

```
         0093 C5  0B   91 0244   779          CMPB    #CDF_WRITEDATA,UCB$B_CEX(R5) ;WRITE DATA FUNCTION?
                  1C   13 0249   780          BEQL    RETRY                       ;IF EQL YES
         0093 C5  0D   91 024B   781          CMPB    #CDF_WRITEHEAD,UCB$B_CEX(R5) ;WRITE HEADER FUNCTION?
                  15   13 0250   782          BEQL    RETRY                       ;IF EQL YES
         51  2820 8F   B3 0252   783          BITW    #RK_CS1_M_CTO!-             ;CONTROLLER TIMEOUT OR,
                          0257   784                  RK_CS1_M_DPPE!-             ;DATAPATH PURGE ERROR OR,
                          0257   785                  RK_CS1_M_SPAR,R1            ;SERIAL BUS PARITY ERROR?
                  0E   12 0257   786          BNEQ    RETRY                       ;IF NEQ YES
         52  E800 8F   B3 0259   787          BITW    #RK_CS2_M_DLT!-             ;DATA LATE OR,
                          025E   788                  RK_CS2_M_UPE!-              ;UNIBUS PARITY ERROR OR,
                          025E   789                  RK_CS2_M_NEM!-              ;NONEXISTENT MEMORY OR,
                          025E   790                  RK_CS2_M_WCE,R2             ;WRITE CHECK ERROR?
                  07   12 025E   791          BNEQ    RETRY                       ;IF NEQ YES
         53  200A 8F   B3 0260   792          BITW    #RK_ER_M_DRPAR!-            ;DRIVE PARITY ERROR OR,
                          0265   793                  RK_ER_M_OPI!-               ;OPERATION INCOMPLETE OR,
                          0265   794                  RK_ER_M_SKI,R3              ;SEEK INCOMPLETE?
                  03   13 0265   795          BEQL    ECC                         ;IF EQL NO
                          0267   796 RETRY:                                       ;
               00C5    31 0267   797          BRW     RETRYERR                    ;RETRIABLE ERROR
                          026A   798
                          026A   799 ;
                          026A   800 ; ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
                          026A   801 ;
                          026A   802
                          026A   803 ECC:                                         ;ECC CORRECTION
         50   00C0 C5  3C 026A   804          MOVZWL  UCB$W_BCR(R5),R0            ;GET NEGATIVE NUMBER OF BYTES REMAINING
           50   7E A5  A0 026F   805          ADDW    UCB$W_BCNT(R5),R0           ;CALCULATE NUMBER OF BYTES TRANSFERED
                51 50  D0 0273   806          MOVL    R0,R1                       ;COPY NUMBER OF BYTES TRANSFERED
                  5F   13 0276   807          BEQL    OFF                         ;IF EQL NONE - PERFORM OFFSET RECOVERY
           02 53  08   E0 0278   808          BBS     #RK_ER_V_HVRC,R3,10$        ;IF SET, HEADER VRC ERROR
                  50   D7 027C   809          DECL    R0                          ;SET TO TRUNCATE LAST BLOCK TRANSFERED
         50  01FF 8F   AA 027E   810 10$:     BICW    #^X1FF,R0                   ;TRUNCATE RESIDUAL BYTES TRANSFERED
         53  1140 8F   B3 0283   811          BITW    #RK_ER_M_DTE!-              ;DRIVE TIMING ERROR OR,
                          0288   812                  RK_ER_M_ECH!-               ;ECC HARD ERROR OR,
                          0288   813                  RK_ER_M_HVRC,R3             ;HEADER VRC ERROR?
                  4D   12 0288   814          BNEQ    OFF                         ;IF NEQ YES - PERFORM OFFSET RECOVERY
               7E 52   7D 028A   815          MOVQ    R2,-(SP)                    ; Save work registers.
   52   00C6 C5  0B 00 EA 028D   816          FFS     #0,#11,UCB$W_EC2(R5),R2     ; Find the first error bit in the ECC
                          0294   817                                              ;  pattern.
         53   0A   52  C3 0294   818          SUBL3   R2,#10,R3                   ; Get the number of error bits
                          0298   819                                              ;  remaining in the pattern.
                  09   15 0298   820          BLEQ    20$                         ; Branch if no other bits in pattern.
                  52   D6 029A   821          INCL    R2                          ; Point ot next bit in pattern.
   52   00C6 C5  53 52 EF 029C   822          EXTZV   R2,R3,UCB$W_EC2(R5),R2      ; Is there more than one error bit set?
                  0C   BA 02A3   823 20$:     POPR    #^M<R3,R2>                  ; Restore work registers without
                          02A5   824                                              ;  affecting flags.
                  2B   1A 02A5   825          BGTRU   DEFER_ECC                   ; If more than one error bit set, don't
                          02A7   826                                              ;  apply ECC correction.
                          02A7   827 ;
                          02A7   828 ; APPLY_ECC -
                          02A7   829 ;
                          02A7   830 ; Apply ECC correction to correct a single bit error.
                          02A7   831 ;
                          02A7   832
                          02A7   833 APPLY_ECC:
              7E  51   D0 02A7   834          MOVL    R1,-(SP)                    ; Save total bytes transfered, inc. ECC.
         00000000'GF   16 02AA   835          JSB     G^IOC$APPLYECC              ; Apply ECC correction.
```

```
            50 8ED0  02B0              POPL    R0                          ; Retrieve transfered byte count.
   00000000'GF   16  02B3              JSB     G^IOC$UPDATRANSP            ; Update transfer parameters.
       00CA C5   94  02B9              CLRB    UCB$B_OFFNDX(R5)            ; Reset offset table index.
                     02BD              EXFUNCH 30$,F_RETCENTER             ; Return to centerline.
         7E A5   B5  02C4              TSTW    UCB$W_BCNT(R5)              ; Any more to transfer?
            03   13  02C7              BEQL    20$                         ; If EQL no.
          FF34   31  02C9              BRW     TRANSFR                     ; Transfer next segment.
          FF3A   31  02CC      20$:    BRW     DATACHECK                   ; Check for write check.
                     02CF
          0079   31  02CF      30$:    BRW     FATALERR                    ; Branch to fatal error routine.
                     02D2
                     02D2    ;
                     02D2    ; DEFER_ECC -
                     02D2    ;
                     02D2    ; Don't apply ECC correction for multiple bit errors unless the error cannot
                     02D2    ; be recovered with offset retries.
                     02D2    ;
                     02D2
                     02D2    DEFER_ECC:
            04   88  02D2              BISB    #DM_M_ECC_DEFER,-           ; Set flag to indicate that ECC
       00C9 C5       02D4                      UCB$W_OFFSET+1(R5)          ;  can be used if offset recovery fails.
                     02D7
                     02D7    ;
                     02D7    ; OFF - OFFSET RECOVERY
                     02D7    ;
                     02D7    ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER VRC, OR ECC HARD
                     02D7    ; ERROR IS DETECTED ON A READ FUNCTION.
                     02D7    ;
                     02D7
                     02D7    OFF:                                              ;OFFSET RECOVERY
    00FA C5    01    88  02D7          BISB    #DM_IND_M_OF,UCB$B_DM_IND(R5)   ;SET OFFSET FLAG
               50    D5  02DC          TSTL    R0                              ;ANY GOOD DATA TRANSFERED?
               0C    13  02DE          BEQL    10$                             ;IF EQL NO
                     02E0
                     02E0    ;
                     02E0    ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
                     02E0    ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
                     02E0    ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
                     02E0    ; POINT OF ERROR.
                     02E0    ;
                     02E0
   00000000'GF   16  02E0              JSB     G^IOC$UPDATRANSP            ;UPDATE TRANSFER PARAMETERS
       00CA C5   94  02E6              CLRB    UCB$B_OFFNDX(R5)            ;RESET OFFSET TABLE INDEX
            0B   11  02EA              BRB     20$                         ;
                     02EC
                     02EC    ;
                     02EC    ; NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
                     02EC    ;
                     02EC
       00CB C5   97  02EC      10$:    DECB    UCB$B_OFFRTC(R5)            ; Change current offset?
            31   12  02F0              BNEQ    50$                         ; If NEQ no.
    00CB C5 02   90  02F2              MOVB    #2,UCB$B_OFFRTC(R5)         ; Set offset retry count.
       00CA C5   96  02F7      20$:    INCB    UCB$B_OFFNDX(R5)            ; Update offset table index.
 50    00CA C5   9A  02FB              MOVZBL  UCB$B_OFFNDX(R5),R0         ; Get next offset table index.
       50   08   91  0300              CMPB    #OFFSIZ,R0                  ; All offsets tried?
            08   12  0303              BNEQ    30$                         ; Branch if not.
            02   E4  0305              BBSC    #DM_V_ECC_DEFER,-           ; Correct the error with ECC if we can.
            00C9 C5        0307   893                        UCB$W_OFFSET+1(R5),-      ;
              9C           030A   894                        APPLY_ECC                 ;
              39     11    030B   895           BRB          OFFSETERR                 ; Otherwise, fatal error.
   00C8 C5   FD45 CF40  90 030D   896 30$:      MOVB         OFFTAB-1[R0],-            ; Set next offset value.
                           0315   897                        UCB$W_OFFSET(R5)          ;
              05     12    0315   898           BNEQ         40$                       ; If NEQ not offset zero.
        00CB C5      10  90 0317  899           MOVB         #16,UCB$B_OFFRTC(R5)      ; Set offset retry count.
                           031C   900 40$:      EXFUNCH FATALERR,F_OFFSET              ; Offset to next position.
              01     E0    0323   901 50$:      BBS          #DM_V_DCK,-               ; If set, data check function.
         03 00C9 C5        0325   902                        UCB$W_OFFSET+1(R5),60$    ;
            FED4     31    0329   903           BRW          TRANSFR                   ; Try function again.
            FF08     31    032C   904 60$:      BRW          CHECKRETRY                ; Try data check again.
                           032F   905
                           032F   906 ;
                           032F   907 ; RETRIABLE ERROR
                           032F   908 ;
                           032F   909
                           032F   910 RETRYERR:                                           ;RETRIABLE ERROR
            0080 C5   97   032F   911           DECB    UCB$B_ERTCNT(R5)                    ;ANY RETRIES LEFT?
              16      13   0333   912           BEQL    FATALERR                            ;IF EQL NO
      53    2002 8F   B3   0335   913           BITW    #RK_ER_M_OPI!-                      ;OPERATION INCOMPLETE OR,
                           033A   914                   RK_ER_M_SKI,R3                      ;SEEK INCOMPLETE?
              07      13   033A   915           BEQL    10$                                 ;IF EQL NO
                           033C   916           EXFUNCH FATALERR,F_RECAL                    ;RECALIBRATE DRIVE
            FE38      31   0343   917 10$:      BRW     FDISPATCH                           ;
                           0346   918
                           0346   919 ;
                           0346   920 ; ALL OFFSETS TRIED - RETRIEVE FINAL TRANSFER STATUS
                           0346   921 ;
                           0346   922
                           0346   923 OFFSETERR:                                          ;OFFSET RECOVERY ERROR
      51    00CE C5   3C   0346   924           MOVZWL  UCB$W_DM_CS1(R5),R1                 ;RETRIEVE CONTROL STATUS REGISTER 1
                           034B   925
                           034B   926 ;
                           034B   927 ; FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
                           034B   928 ; INHIBITED, OR FINAL OFFSET TRIED
                           034B   929 ;
                           034B   930
                           034B   931 FATALERR:                                           ;FATAL ERROR - SET STATUS
      50    0254 8F   3C   034B   932           MOVZWL  #SS$_VOLINV,R0                      ;SET VOLUME INVALID STATUS
   6B 00D8 C5   06    E1   0350   933           BBC     #RK_DS_V_VV,UCB$W_DM_DS(R5),FUNCXT ;IF CLR, VOLUME INVALID
      50    023C 8F   3C   0356   934           MOVZWL  #SS$_UNSAFE,R0                      ;SET DRIVE UNSAFE STATUS
        62  53   0E   E0   0358   935           BBS     #RK_ER_V_UNS,R3,FUNCXT              ;IF SET, DRIVE UNSAFE
      50    00BC 8F   3C   035F   936           MOVZWL  #SS$_FORMAT,R0                      ;SET FORMAT ERROR STATUS
            53   30   B3   0364   937           BITW    #RK_ER_M_DTYE!-                     ;DRIVE TYPE ERROR OR,
                           0367   938                   RK_ER_M_FMTE,R3                     ;FORMAT ERROR?
                 5B   12   0367   939           BNEQ    FUNCXT                              ;IF NEQ YES
      50    025C 8F   3C   0369   940           MOVZWL  #SS$_WRITLCK,R0                     ;SET WRITE LOCK ERROR
        4F  53   08   E0   036E   941           BBS     #RK_ER_V_WLE,R3,FUNCXT              ;IF SET, WRITE LOCK ERROR
      50    0134 8F   3C   0372   942           MOVZWL  #SS$_IVADDR,R0                      ;SET INVALID DISK ADDRESS STATUS
      53    0600 8F   B3   0377   943           BITW    #RK_ER_M_COE!-                      ;CYLINDER ADDRESS OVERFLOW OR,
                           037C   944                   RK_ER_M_IDAE,R3                     ;INVALID DISK ADDRESS ERROR?
                 43   12   037C   945           BNEQ    FUNCXT                              ;IF NEQ YES
      50    008C 8F   3C   037E   946           MOVZWL  #SS$_DRVERR,R0                      ;SET DRIVE ERROR STATUS
      53    3007 8F   B3   0383   947           BITW    #RK_ER_M_DTE!-                      ;DRIVE TIMING ERROR OR,
                           0388   948                   RK_ER_M_ILF!-                       ;ILLEGAL FUNCTION OR,
                           0388   949                   RK_ER_M_NXF!-                       ;NONEXECUTABLE FUNCTION OR,
                        0388                    RK_ER_M_OPI!-                   ;OPERATION INCOMPLETE OR,
                        0388                    RK_ER_M_SKI,R3                  ;SEEK INCOMPLETE?
               37   12  0388            BNEQ    FUNCXT                          ;IF NEQ YES
    50   01F4  8F   3C  038A            MOVZWL  #SS$_PARITY,R0                  ; Set parity error status.
    53   81C0  8F   B3  038F            BITW    #RK_ER_M_BSE!-                  ; Bad sector error or,
                        0394                    RK_ER_M_DCK!-                   ;   data check error or,
                        0394                    RK_ER_M_ECH!-                   ;   ECC hard error or,
                        0394                    RK_ER_M_HVRC,R3                 ;   header VRC error?
               2B   12  0394            BNEQ    FUNCXT                          ; If NEQ yes
    50   0054  8F   3C  0396            MOVZWL  #SS$_CTRLERR,R0                 ; Set fatal controller error status.
      22 53    03   E0  0398            BBS     #RK_ER_V_DRPAR,R3,-             ; Branch if drive parity error.
                        039F                    FUNCXT                          ;
    51   2020  8F   B3  039F            BITW    #RK_CS1_M_DPPE!-                ;DATAPATH PURGE ERROR OR,
                        03A4                    RK_CS1_M_SPAR,R1                ;SERIAL BUS PARITY ERROR?
               1B   12  03A4            BNEQ    FUNCXT                          ;IF NEQ YES
      17 52    0D   E0  03A6            BBS     #RK_CS2_V_UPE,R2,FUNCXT ;IF SET, UNIBUS PARITY ERROR
    50   005C  8F   3C  03AA            MOVZWL  #SS$_DATACHECK,R0               ;SET DATA CHECK ERROR STATUS
      0E 52    0E   E0  03AF            BBS     #RK_CS2_V_WCE,R2,FUNCXT ;IF SET, WRITE CHECK ERROR
    50   01C4  8F   3C  03B3            MOVZWL  #SS$_NONEXDRV,R0                ;SET NONEXISTENT DRIVE STATUS
      05 52    0C   E0  03B8            BBS     #RK_CS2_V_NED,R2,FUNCXT ;IF SET, NONEXISTENT DRIVE
    50   0054  8F   3C  03BC            MOVZWL  #SS$_CTRLERR,R0                 ;SET CONTROLLER ERROR STATUS
                        03C1
                        03C1    ;
                        03C1    ; FUNCTION COMPLETION COMMON EXIT
                        03C1    ;
                        03C1
                        03C1    FUNCXT:                                         ;FUNCTION EXIT
   00FB C5     50   D0  03C1            MOVL    R0,UCB$L_DM_FRS(R5)             ;SAVE FINAL REQUEST STATUS ACROSS RECAL
    00000000'GF     16  03C6            JSB     G^IOC$DIAGBUFILL                ;FILL DIAGNOSTIC BUFFER IF PRESENT
   0092 C5     0A   91  03CC            CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
               14   1A  03D1            BGTRU   10$                             ;IF GTRU YES
   0092 C5     0F   91  03D3            CMPB    #CDF_AVAILABLE, UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
               0D   13  03D8            BEQL    10$                             ;IF EQL YES
     53     58 A5   D0  03DA            MOVL    UCB$L_IRP(R5),R3                ;RETRIEVE ADDRESS OF IRP
          00C0 C5   A1  03DE            ADDW3   UCB$W_BCR(R5),-                 ;CALCULATE AND SAVE BYTES TRANSFERRED
 00FD C5    32 A3       03E2                    IRP$W_BCNT(R3),UCB$L_DM_FRS+2(R5)  ;...
 0C 00FA C5    00   E5  03E7    10$:    BBCC    #DM_IND_V_OF,UCB$B_DM_IND(R5),20$  ;IF CLEAR, NOT IN OFFSET MODE
   0080 C5     01   90  03ED            MOVB    #1,UCB$B_ERTCNT(R5)             ; Set error retry count to 1 to
                        03F2                                                    ;   prevent a timeout on the following
                        03F2                                                    ;   RECAL from decrementing the count
                        03F2                                                    ;   to a negative number and thereby
                        03F2                                                    ;   triggering a semi-infinite loop.
                        03F2            EXFUNCH 20$,F_RECAL                     ;RECALIBRATE DRIVE
               51   D4  03F9    20$:    CLRL    R1                              ;CLEAR SECOND STATUS LONGWORD
    50   00FB  C5   D0  03FB            MOVL    UCB$L_DM_FRS(R5),R0             ;RETRIEVE FINAL REQUEST STATUS
                        0400            REQCOM                                  ;COMPLETE REQUEST
```

```
0406            .SBTTL  RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION
0406    ;
0406    ; FEXH - RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION (HIGH PRIORITY)
0406    ; FEXL - RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION (LOW PRIORITY)
0406    ;
0406    ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
0406    ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
0406    ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
0406    ; INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. THE RETURN ADDRESS
0406    ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
0406    ; RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
0406    ;
0406    ; INPUTS:
0406    ;
0406    ;       R3 = FUNCTION TABLE DISPATCH INDEX.
0406    ;       R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
0406    ;       R5 = DEVICE UNIT UCB ADDRESS.
0406    ;
0406    ;       00(SP) = RETURN ADDRESS OF CALLER.
0406    ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
0406    ;
0406    ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
0406    ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
0406    ;
0406    ; OUTPUTS:
0406    ;
0406    ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
0406    ;
0406    ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
0406    ;               OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
0406    ;               ERROR ROUTINE.
0406    ;
0406    ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
0406    ;               ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
0406    ;               INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
0406    ;
0406    ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
0406    ;               OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
0406    ;               IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
0406    ;
0406    ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
0406    ;               DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
0406    ;
0406    ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
0406    ;
0406    ;       IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
0406    ;       THE GENERAL REGISTERS R1, R2, AND R3, AND THE UCB.
0406    ;
0406    ;       R1 = CONTROL STATUS REGISTER 1.
0406    ;       R2 = CONTROL STATUS REGISTER 2.
0406    ;       R3 = ERROR REGISTER.
0406    ;
0406    ;       UCB$W_EC1(R5) = ECC POSITION REGISTER.
0406    ;       UCB$W_EC2(R5) = ECC PATTERN REGISTER.
0406    ;       UCB$W_BCR(R5) = BYTE COUNT REGISTER.
0406    ;
0406
```

```
                                 0406          .ENABL  LSB
                                 0406  FEXH:                                     ;FUNCTION EXECUTOR (HIGH PRIORITY)
         52   00000000'GF    9E  0406          MOVAB   G^IOC$REQPCHANH,R2        ;SET ADDRESS OF REQUEST CHANNEL ROUTINE
                       07    11  040D          BRB     10$
                                 040F  FEXL:                                     ;FUNCTION EXECUTOR (LOW PRIORITY)
         52   00000000'GF    9E  040F          MOVAB   G^IOC$REQPCHANL,R2        ;SET ADDRESS OF REQUEST CHANNEL ROUTINE
                 009C C5   8ED0  0416  10$:    POPL    UCB$L_DPC(R5)             ;SAVE DRIVER PC VALUE
            0093 C5    53    90  0418          MOVB    R3,UCB$B_CEX(R5)          ;SAVE CASE INDEX
                       62    16  0420          JSB     (R2)                      ;REQUEST CHANNEL
              52    54 A5    3C  0422          MOVZWL  UCB$W_UNIT(R5),R2         ;GET DEVICE UNIT NUMBER
                                 0426          CASE    R3,<-                     ;DISPATCH TO PROPER FUNCTION ROUTINE
                                 0426                  IMMED,-                   ;NO OPERATION
                                 0426                  IMMED,-                   ;UNLOAD VOLUME
                                 0426                  POSIT,-                   ;SEEK CYLINDER
                                 0426                  POSIT,-                   ;RECALIBRATE
                                 0426                  IMMED,-                   ;DRIVE CLEAR
                                 0426                  RELES,-                   ;RELEASE DRIVE
                                 0426                  POSIT,-                   ;OFFSET HEADS
                                 0426                  POSIT,-                   ;RETURN TO CENTERLINE
                                 0426                  IMMED,-                   ;PACK ACKNOWLEDGE
                                 0426                  IMMED,-                   ;START SPINDLE
                                 0426                  >                         ;
                     00A1    31  043E          BRW     XFER                      ;TRANSFER FUNCTION
                                 0441          .DSABL  LSB
                                 0441
                                 0441  ;
                                 0441  ; IMMEDIATE FUNCTION EXECUTION
                                 0441  ;
                                 0441  ;       FUNCTIONS INCLUDE:
                                 0441  ;
                                 0441  ;               NO OPERATION,
                                 0441  ;               UNLOAD VOLUME,
                                 0441  ;               DRIVE CLEAR,
                                 0441  ;               RELEASE PORT,
                                 0441  ;               PACK ACKNOWLEDGE, AND
                                 0441  ;               START SPINDLE.
                                 0441  ;
                                 0441  ; INTERRUPTS ARE LOCKED OUT, THE APPROPRIATE FUNCTION IS INITIATED WITH
                                 0441  ; INTERRUPT ENABLE, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
                                 0441  ;
                                 0441  ; THESE FUNCTIONS ALL EXECUTE WITHIN A VERY SHORT TIME (15 US), BUT ARE
                                 0441  ; VERY INFREQUENT AND THEREFORE ARE DONE WITH INTERRUPTS TO AVOID AN EXTRA
                                 0441  ; SET OF REGISTER SAVE LOGIC.
                                 0441  ;
                                 0441
                                 0441  RELES:                                    ;RELEASE PORT
                 52    08    A8  0441          BISW    #RK_CS2_M_RLS,R2          ;SET PORT RELEASE BIT
                                 0444  IMMED:                                    ;IMMEDIATE FUNCTION EXECUTION
                                 0444          DSBINT                            ;DISABLE INTERRUPTS
           03 64 A5    05    E1  044A          BBC     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF CLR, POWER HAS NOT FAILED
                     008A    31  044F          BRW     ENBXIT                    ;ELSE, POWER HAS FAILED
              08 A4    52    B0  0452  10$:    MOVW    R2,RK_CS2(R4)             ;SET UNIT NUMBER
  64    FBDA CF43   00CC C5  A9  0456          BISW3   UCB$W_DM_DTYP(R5),FTAB[R3],RK_CS1(R4) ;EXECUTE FUNCTION
                                 045F          WFIKPCH RLSCHN,#2                 ;WAITFOR INTERRUPT
                                 0469          IOFORK                            ;CREATE FORK PROCESS
                     01A1    31  046F          BRW     RLSCHN                    ;
                                 0472
```

```
                                   0472  ;
                                   0472  ; POSITIONING FUNCTION EXECUTION
                                   0472  ;
                                   0472  ;       FUNCTIONS INCLUDE:
                                   0472  ;
                                   0472  ;               SEEK CYLINDER,
                                   0472  ;               RECALIBRATE,
                                   0472  ;               OFFSET HEADS, AND
                                   0472  ;               RETURN HEADS TO CENTERLINE.
                                   0472  ;
                                   0472  ; THE OFFSET REGISTER AND CYLINDER ADDRESS REGISTERS ARE LOADED, INTERRUPTS
                                   0472  ; ARE LOCKED OUT, AND THE APPROPRIATE POSITIONING FUNCTION IS INITIATED
                                   0472  ; WITHOUT INTERRUPT ENABLE. THE CONTROLLER IS THEN POLLED FOR READY, DEVICE
                                   0472  ; INTERRUPTS ARE ENABLED, AND A WAITFOR INTERRUPT AND RELEASE CHANNEL IS
                                   0472  ; EXECUTED.
                                   0472  ;
                                   0472
                                   0472  POSIT:                                          ;POSITIONING FUNCTION
                                   0472          DSBINT                                  ;DISABLE INTERRUPTS
              5F 64 A5   05    E0  0478          BBS     #UCB$V_POWER,UCB$W_STS(R5),ENBXIT ;IF SET, POWER HAS FAILED
          0E A4   00C8 C5    B0  047D          MOVW    UCB$W_OFFSET(R5),RK_AS(R4)  ;SET OFFSET VALUE
          10 A4   00BE C5    B0  0483          MOVW    UCB$W_DC(R5),RK_DC(R4)      ;SET DESIRED CYLINDER ADDRESS
                  08 A4   52    B0  0489          MOVW    R2,RK_CS2(R4)               ;SET UNIT NUMBER
   64   FBA3 CF43   00CC C5    A9  048D          BISW3   UCB$W_DM_DTYP(R5),FTAB[R3],RK_CS1(R4) ;EXECUTE FUNCTION
                       03C3    30  0496          BSBW    DM_WAIT                     ;WAIT FOR FUNCTION TO COMPLETE
                                   0499          WFIRLCH RETREG,#6                   ;WAITFOR INTERRUPT
                                   04A3  5$:
                                   04A3          DSBINT                              ;DISABLE INTERRUPTS
              22 64 A5   05    E0  04A9          BBS     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF SET, POWER FAILURE
              08 A4   54 A5    B0  04AE          MOVW    UCB$W_UNIT(R5),RK_CS2(R4)   ;SET UNIT NUMBER
        64   01   00CC C5    A9  04B3          BISW3   UCB$W_DM_DTYP(R5),#F_NOP!1,RK_CS1(R4) ;SELECT DRIVE TO GET STATUS
                       03A0    30  04B9          BSBW    DM_WAIT                     ;WAIT FOR CONTROLLER READY
          0A A4   4000 8F    B3  04BC          BITW    #RK_DS_M_DSC,RK_DS(R4)      ;OPERATION COMPLETE?
                         0C    12  04C2          BNEQ    10$                         ;BR IF YES
                                   04C4          WFIKPCH RETREG,#6                   ;WAITFOR INTERRUPT
                         D3    11  04CE          BRB     5$                          ;
                                   04D0  10$:    ENBINT                              ;ENABLE INTERRUPTS
                                   04D3  20$:    IOFORK                              ;CREATE FORK PROCESS
                       013D    31  04D9          BRW     RETREG                      ;
                                   04DC  ENBXIT:                                     ;
                                   04DC          ENBINT                              ;
                       0131    31  04DF          BRW     RLSCHN                      ;
                                   04E2
                                   04E2  ;
                                   04E2  ; TRANSFER FUNCTION EXECUTION
                                   04E2  ;
                                   04E2  ;       FUNCTIONS INCLUDE:
                                   04E2  ;
                                   04E2  ;               WRITE CHECK,
                                   04E2  ;               WRITE DATA,
                                   04E2  ;               WRITE HEADERS,
                                   04E2  ;               READ DATA, AND
                                   04E2  ;               READ HEADER.
                                   04E2  ;
                                   04E2  ; A UNIBUS DATAPATH IS REQUESTED FOLLOWED BY THE APPROPRIATE NUMBER OF MAP
                                   04E2  ; REGISTERS REQUIRED FOR THE TRANSFER. THE TRANSFER PARAMETERS ARE LOADED
                                   04E2  ; INTO THE DEVICE REGISTERS, INTERRUPTS ARE LOCKED OUT, THE FUNCTION IS
```

```
                              04E2  ; INITIATED, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
                              04E2  ;
                              04E2
                              04E2  XFER:                                          ;TRANSFER FUNCTION EXECUTION
                              04E2             REQDPR                              ;REQUEST DATAPATH
                              04E8             REQMPR                              ;REQUEST MAP REGISTERS
                              04EE             LOADUBA                             ;LOAD UNIBUS MAP REGISTERS
            50    7E A5    3C 04F4             MOVZWL  UCB$W_BCNT(R5),R0           ;GET TRANSFER BYTE COUNT
               50    02    C6 04F8             DIVL    #2,R0                       ;CALCULATE TRANSFER WORD COUNT
                              04FB             DSBINT  UCB$B_DIPL(R5)              ;DISABLE DEVICE INTERRUPTS
            02 A4    50    AE 0502             MNEGW   R0,RK_WC(R4)                ;SET TRANSFER WORD COUNT
            50    7C A5    3C 0506             MOVZWL  UCB$W_BOFF(R5),R0           ;GET BYTE OFFSET IN PAGE
            51    24 A5    D0 050A             MOVL    UCB$L_CRB(R5),R1            ;GET ADDRESS OF CRB
   50    07 09    34 A1    F0 050E             INSV    CRB$L_INTD+VEC$W_MAPREG(R1),#9,#7,R0 ;INSERT HIGH 7 BITS OF ADDRESS
            04 A4    50    B0 0514             MOVW    R0,RK_BA(R4)                ;SET BUFFER ADDRESS
   50    34 A1 02    07    EF 0518             EXTZV   #7,#2,CRB$L_INTD+VEC$W_MAPREG(R1),R0 ;GET MEMORY EXTENSION BITS
            50    50 08    78 051E             ASHL    #8,R0,R0                    ;SHIFT LEFT ONE BYTE
       50    FB11 CF43     A8 0522             BISW    FTAB[R3],R0                 ;MERGE FUNCTION CODE
         06 A4    00BC C5  B0 0528             MOVW    UCB$W_DA(R5),RK_DA(R4)      ;SET DESIRED TRACK AND SECTOR ADDRESS
         10 A4    00BE C5  B0 052E             MOVW    UCB$W_DC(R5),RK_DC(R4)      ;SET DESIRED CYLINDER ADDRESS
          08 A4      54 A5 B0 0534             MOVW    UCB$W_UNIT(R5),RK_CS2(R4)   ;SET UNIT NUMBER
                              0539             SETIPL  #31                         ;DISABLE INTERRUPTS
          06 64 A5    05   E1 053C             BBC     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF CLR, NO POWER FAILURE
                              0541             ENBINT                              ;ENABLE INTERRUPTS
                    00BC   31 0544             BRW     60$                         ;
      64    50    00CC C5  A9 0547  10$:       BISW3   UCB$W_DM_DTYP(R5),R0,RK_CS1(R4) ;EXECUTE FUNCTION
                              054D             WFIKPCH 60$,#6                      ;WAITFOR INTERRUPT AND KEEP CHANNEL
                              0557             IOFORK                              ;CREATE FORK PROCESS
                              055D             PURDPR                              ;PURGE DATAPATH, CHECK/CLEAR ERRORS
                   16 50   E8 0563             BLBS    R0,20$                      ;BRANCH IF NO DATAPATH ERROR
      00CE C5    8020 8F   A8 0566             BISW    #RK_CS1_M_CERR!-            ;SET CONTROLLER ERROR AND,
                              056D                     RK_CS1_M_DPPE,UCB$W_DM_CS1(R5) ;DATAPATH PURGE ERROR
                 00D6 C5   B4 056D             CLRW    UCB$W_DM_CS2(R5)            ;CLEAR CONTROL STATUS REGISTER 2
                 00DA C5   B4 0571             CLRW    UCB$W_DM_ER(R5)             ;CLEAR ERROR REGISTER
      00D8 C5    FFBF 8F   AA 0575             BICW    #^C<RK_DS_M_VV>,UCB$W_DM_DS(R5) ;CLEAR ALL BUT VOLUME VALID
        05 00CE C5    0F   E0 057C  20$:       BBS     #RK_CST_V_CERR,UCB$W_DM_CS1(R5),30$ ;IF SET , DEVICE ERRORS
          41 68 A5    01   E1 0582             BBC     #UCB$V_DIAGBUF,UCB$W_DEVSTS(R5),40$ ;IF CLR, NO DIAGNOSTIC BUFFER
                      00   EF 0587  30$:       EXTZV   #VEC$V_DATAPATH,-           ;EXTRACT DATAPATH #
                      05      0589                     #VEC$S_DATAPATH,-           ; FROM CRB AND SAVE IT
                   37 A3      058A                     CRB$L_INTD+VEC$B_DATAPATH(R3),-
                 00E6 C5      058C                     UCB$W_DM_DPN(R5)            ;
          00E8 C5    51    D0 058F             MOVL    R1,UCB$L_DM_DPR(R5)         ;SAVE DATAPATH REGISTER CONTENTS
   50    00D2 C5  07    09 EF 0594             EXTZV   #9,#7,UCB$W_DM_BA(R5),R0    ;GET LOW BITS OF FINAL
                              059B                                                 ; MAP REGISTER NUMBER
   50    02    07    00CF C5 F0 059B           INSV    UCB$W_DM_CS1+1(R5),#7,#2,R0 ;INSERT HIGH BITS OF FINAL MAP REGISTER
            50    01EF 8F  B1 05A2             CMPW    #495,R0                     ;LEGAL MAP REGISTER NUMBER?
                      05   18 05A7             BGEQ    35$                         ;IF GEQ YES
            50    01EF 8F  3C 05A9             MOVZWL  #495,R0                     ;RESTRICT MAP REGISTER NUMBER
       00EC C5      6240   D0 05AE  35$:       MOVL    (R2)[R0],UCB$L_DM_FMPR(R5)  ;SAVE FINAL MAP REGISTER
                 00F0 C5   D4 05B4             CLRL    UCB$L_DM_PMPR(R5)           ;CLEAR PREVIOUS MAP REGISTER CONTENTS
                      50   D7 05B8             DECL    R0                          ;CALCULATE PREVIOUS MAP REGISTER NUMBER
                0F    00   EC 05BA             CMPV    #VEC$V_MAPREG,#VEC$S_MAPREG,- ;ANY PREVIOUS MAP REGISTER?
            50    34 A3       05BD                     CRB$L_INTD+VEC$W_MAPREG(R3),R0 ;
                      06   14 05C0             BGTR    40$                         ;IF GTR NO
       00F0 C5      6240   D0 05C2             MOVL    (R2)[R0],UCB$L_DM_PMPR(R5)  ;SAVE PREVIOUS MAP REGISTER
   00C0 C5    00D0 C5 02   A5 05C8  40$:       MULW3   #2,UCB$W_DM_WC(R5),UCB$W_BCR(R5) ;CONVERT WD TO BYTE COUNT
        2D 00CE C5    0F   E0 05D0             BBS     #RK_CS1_V_CERR,UCB$W_DM_CS1(R5),60$ ;IF SET, DEVICE ERRORS
      0093 C5   0E   91  05D6          CMPB    #CDF_READHEAD,UCB$B_CEX(R5) ;READ HEADER FUNCTION?
                26   12  05DB          BNEQ    60$                       ;IF NEQ NO
             78 A5   DD  05DD          PUSHL   UCB$L_SVAPTE(R5)          ;SAVE ADDRESS OF PTE
       51   00F4 C5  9E  05E0          MOVAB   UCB$W_DM_DB(R5),R1        ;SET ADDRESS OF INTERNAL BUFFER
                52   06   D0 05E5      MOVL    #6,R2                     ;SET NUMBER OF BYTES TO MOVE
          7E A5 52   B1  05E8          CMPW    R2,UCB$W_BCNT(R5)         ;ROOM FOR FULL HEADER?
                04   1F  05EC          BLSSU   50$                       ;IF LSSU YES
          52   7E A5 3C  05EE          MOVZWL  UCB$W_BCNT(R5),R2         ;SET LENGTH OF PARTIAL HEADER
 00C0 C5   52   7E A5 A3 05F2  50$:    SUBW3   UCB$W_BCNT(R5),R2,UCB$W_BCR(R5) ;CALCULATE TRANSFER BYTE COUNT
       00000000'GF   16  05F9          JSB     G^IOC$MOVTOUSER           ;MOVE HEADER TO USER BUFFER
             78 A5 8ED0  05FF          POPL    UCB$L_SVAPTE(R5)          ;RESTORE ADDRESS OF PTE
                         0603  60$:    SETIPL  UCB$B_FIPL(R5)            ;INSURE PROPER IPL FOR RELEASE
                         0607          RELDPR                            ;RELEASE DATA PATH
                         060D          RELMPR                            ;RELEASE MAP REGISTERS
                         0613  RLSCHN:                                   ;
                         0613          RELCHAN                           ;RELEASE CHANNEL
                         0619
                         0619  ;
                         0619  ; RETURN REGISTERS
                         0619  ;
                         0619
                         0619          .ENABL  LSB
                         0619  RETREG:                                   ;RETURN FINAL DEVICE REGISTERS
       51    00CE C5  3C 0619          MOVZWL  UCB$W_DM_CS1(R5),R1       ;RETRIEVE CONTROL STATUS REGISTER 1
       52    00D6 C5  3C 061E          MOVZWL  UCB$W_DM_CS2(R5),R2       ;RETRIEVE CONTROL STATUS REGISTER 2
       53    00DA C5  3C 0623          MOVZWL  UCB$W_DM_ER(R5),R3        ;RETRIEVE ERROR REGISTER
    64 A5    0060 8F  B3 0628          BITW    #UCB$M_POWER!-            ;POWER FAIL OR DEVICE TIMEOUT?
                         062E                  UCB$M_TIMOUT,UCB$W_STS(R5) ;
                64   12  062E          BNEQ    40$                       ;IF NEQ YES - SPECIAL CONDITION
      1D 51     0F   E0  0630          BBS     #RK_CS1_V_CERR,R1,5$      ;IF SET, ERROR OCCURED
                08   91  0634          CMPB    #CDF_PACKACK,-            ; DID WE EXECUTE A PACK ACKNOWLEDGE
           0093 C5       0636                  UCB$B_CEX(R5)             ;  FUNCTION?
                51   12  0639          BNEQ    30$                       ; BRANCH IF NOT.
           0080 8F   B3  063B          BITW    #RK_DS_M_DRDY,-           ; DRIVE READY BIT SET?
           00D8 C5       063F                  UCB$W_DM_DS(R5)           ;
                48   12  0642          BNEQ    30$                       ; BRANCH IF SO.
           0040 8F   AA  0644          BICW    #RK_DS_M_VV,-             ; FORCE VOLUME VALID BIT TO REFLECT TRUE
           00D8 C5       0648                  UCB$W_DM_DS(R5)           ;  STATUS.
    64 A5  0800 8F   AA  064B          BICW    #UCB$M_VALID, UCB$W_STS(R5) ;MARK VOLUME ''INVALID''
     0093 C5    0A   91  0651  5$:     CMPB    #CDF_WRITECHECK,UCB$B_CEX(R5) ;DRIVE RELATED FUNCTION?
                06   1B  0656          BLEQU   10$                       ;IF LEQU NO
  00C0 C5    7E A5   AE  0658          MNEGW   UCB$W_BCNT(R5),UCB$W_BCR(R5) ;RESET BYTE COUNT - NO TRANSFER
                         065E  10$:
      00000000'GF    16  065E          JSB     G^ERL$DEVICERR            ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
   6B 009A C5   0F   E0  0664          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),70$ ;IF SET, RETRY INHIBITED
     52    1700 8F   B3  066A          BITW    #RK_CS2_M_MDS!-           ;MULTIPLE DRIVE SELECT OR,
                         066F                  RK_CS2_M_NED!-            ;NONEXISTENT DISK OR,
                         066F                  RK_CS2_M_PGE!-            ;PROGRAMMING ERROR OR,
                         066F                  RK_CS2_M_UFE,R2           ;UNIT FIELD ERROR?
                64   12  066F          BNEQ    70$                       ;IF NEQ YES
     53    0EB5 8F   B3  0671          BITW    #RK_ER_M_BSE!-            ;BAD SECTOR ERROR OR,
                         0676                  RK_ER_M_COE!-             ;CYLINDER ADDRESS OVERFLOW OR,
                         0676                  RK_ER_M_DTYE!-            ;DRIVE TYPE ERROR OR,
                         0676                  RK_ER_M_FMTE!-            ;FORMAT ERROR OR,
                         0676                  RK_ER_M_IDAE!-            ;INVALID DISK ADDRESS ERROR OR,
                         0676                  RK_ER_M_ILF!-             ;ILLEGAL FUNCTION OR,
                         0676                  RK_ER_M_NXF!-             ;NONEXECUTABLE FUNCTION OR,
                                0676  1282                    RK_ER_M_WLE,R3             ;WRITE LOCK ERROR?
                   5D     12    0676  1283           BNEQ     70$                        ;IF NEQ YES
             1A 53        0E    E0    0678  1284     BBS      #RK_ER_V_UNS,R3,50$        ; Branch if drive is unsafe.
       53 00D8 C5     06    E1  067C  1285           BBC      #RK_DS_V_VV,UCB$W_DM_DS(R5),70$ ;IF CLR, VOLUME INVALID
                                0682  1286
                                0682  1287  ;
                                0682  1288  ; RETRIABLE ERROR EXIT
                                0682  1289  ;
                                0682  1290
          7E    009C D5   98    0682  1291  20$:     CVTBL    @UCB$L_DPC(R5),-(SP)       ;GET BRANCH DISPLACEMENT
          009C C5   8E    C0    0687  1292           ADDL     (SP)+,UCB$L_DPC(R5)        ;CALCULATE RETURN ADDRESS - 1
                009C C5   D6    068C  1293  30$:     INCL     UCB$L_DPC(R5)              ;ADJUST TO CORRECT RETURN ADDRESS
                009C D5   17    0690  1294           JMP      @UCB$L_DPC(R5)             ;RETURN TO DRIVER
                                0694  1295
                   42     11    0694  1296  40$:     BRB      80$                        ;
                                0696  1297
                                0696  1298  ;
                                0696  1299  ; Check for unsafe condition and attempt to clear it.
                                0696  1300  ;
                                0696  1301
                                0696  1302  50$:     DSBINT                              ; Disable interupts.
                   05     E1    069C  1303           BBC      #UCB$V_POWER,-             ; Branch if no power failure occured.
             03 64 A5           069E  1304                    UCB$W_STS(R5),60$          ;
                 FE38     31    06A1  1305           BRW      ENBXIT                     ; Otherwise, enable interupts and
                                06A4  1306                                               ;  go process error.
          00CC C5         A9    06A4  1307  60$:     BISW3    UCB$W_DM_DTYP(R5),-        ; Attempt to clear unsafe condition.
          64      05            06A8  1308           #F_DRVCLR!1,RK_CS1(R4)
                                06AA  1309           TIMEWAIT -                          ; Wait for ten microseconds or until
                                06AA  1310                    TIME = #1,-                ;  unsafe condition clears.
                                06AA  1311                    BITVAL = #RK_CS1_M_CERR,-  ;
                                06AA  1312                    SOURCE = RK_CS1(R4),-      ;
                                06AA  1313                    CONTEXT = W,-              ;
                                06AA  1314                    SENSE = .FALSE.
                                06CF  1315           ENBINT                              ; Enable interupts.
             AD 50        E8    06D2  1316           BLBS     R0,20$                     ; Branch if drive is no longer unsafe.
                                06D5  1317
                                06D5  1318  ;
                                06D5  1319  ; FATAL CONTROLLER OR DRIVE ERROR EXIT
                                06D5  1320  ;
                                06D5  1321
                 FC73     31    06D5  1322  70$:     BRW      FATALERR                   ;
                                06D8  1323
                                06D8  1324  ;
                                06D8  1325  ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
                                06D8  1326  ;
                                06D8  1327
       54 64 A5   05      E4    06D8  1328  80$:     BBSC     #UCB$V_POWER,UCB$W_STS(R5),110$ ;IF SET, POWER FAILURE
                                06DD  1329
                                06DD  1330  ;
                                06DD  1331  ; DEVICE TIME OUT
                                06DD  1332  ;
                                06DD  1333
        00000000'GF       16    06DD  1334           JSB      G^ERL$DEVICTMO             ;LOG DEVICE TIME OUT
          53    24 A5     D0    06E3  1335           MOVL     UCB$L_CRB(R5),R3           ;GET ADDRESS OF CRB
          53    2C A3     D0    06E7  1336           MOVL     CRB$L_INTD+VEC$L_IDB(R3),R3 ;GET ADDRESS OF IDB
          04 A3     55    D1    06EB  1337           CMPL     R5,IDB$L_OWNER(R3)         ;DEVICE OWN CONTROLLER?
                    08    12    06EF  1338           BNEQ     90$                        ;IF NEQ NO
```

```
      08 A4    20   B0  06F1              MOVW    #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR ENTIRE RK611 SUBSYSTEM
      64    40 8F   9B  06F5              MOVZBW  #RK_CS1_M_IE,RK_CS1(R4) ;ENABLE DEVICE INTERRUPTS
                        06F9      90$:    SETIPL  UCB$B_FIPL(R5)            ;LOWER TO FORK LEVEL
                        06FD
               07   E0  06FD              BBS     #RK_DS_V_DRDY,-
       07 00D8 C5       06FF                      UCB$W_DM_DS(R5),100$      ;BR. IF DEVICE READY
   50     01A4 8F   3C  0703              MOVZWL  #SS$_MEDOFL,R0            ;RETURN MEDIUM OFFLINE ERROR
               1A   11  0708              BRB     RESETXFR                  ;EXIT WITHOUT RETRY
                        070A
   50     022C 8F   3C  070A      100$:   MOVZWL  #SS$_TIMEOUT,R0           ;SET DEVICE TIMEOUT STATUS
          0080 C5   97  070F              DECB    UCB$B_ERTCNT(R5)          ;ANY ERROR RETRIES REMAINING?
               0F   13  0713              BEQL    RESETXFR                  ;IF EQL NO
                        0715              RELCHAN                           ;RELEASE CHANNEL IF OWNED
 64 A5    0040 8F   AA  071B              BICW    #UCB$M_TIMOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
             FA5A   31  0721              BRW     FDISPATCH                 ;
                        0724
                        0724      ;
                        0724      ; RESET TRANSFER BYTE COUNT TO ZERO
                        0724      ;
                        0724
                        0724      RESETXFR:                                 ;
      53    58 A5   D0  0724              MOVL    UCB$L_IRP(R5),R3          ;RETRIEVE ADDRESS OF I/O PACKET
 00C0 C5    32 A3   AE  0728              MNEGW   IRP$W_BCNT(R3),UCB$W_BCR(R5) ;RESET TRANSFER BYTE COUNT
             FC90   31  072E              BRW     FUNCXT                    ;
                        0731
                        0731      ;
                        0731      ; POWER FAILURE
                        0731      ;
                        0731
                        0731      110$:   RELCHAN                           ;RELEASE CHANNEL IF OWNED
      53    58 A5   D0  0737              MOVL    UCB$L_IRP(R5),R3          ;RETRIEVE ADDRESS OF I/O PACKET
   78 A5    2C A3   7D  073B              MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
             F9CE   31  0740              BRW     DM_STARTIO                ;
                        0743              .DSABL  LSB
```

```
                         0743          .SBTTL  RK611-RK06/RK07 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                         0743     ;
                         0743     ; DM_DTYPE - RK611-RK06/RK07 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                         0743     ;
                         0743     ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
                         0743     ; SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
                         0743     ; SET UNIT PARAMETERS.
                         0743     ;
                         0743     ; INPUTS:
                         0743     ;
                         0743     ;       R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
                         0743     ;       R5 = DEVICE UNIT UCB ADDRESS.
                         0743     ;
                         0743     ; OUTPUTS:
                         0743     ;
                         0743     ;       THE DRIVE STATUS REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
                         0743     ;
                         0743     ;
                         0743   DM_DTYPE:                                       ;CLASSIFY DRIVE TYPE AND SET UNIT PARAMETERS
              01     90  0743           MOVB    S^#DT$_RK06,-
           41 A5         0745            UCB$B_DEVTYPE(R5)      ;SET RK06 DEVICE TYPE
 46 A5   019B 8F     B0  0747           MOVW    #411,UCB$W_CYLINDERS(R5);SET NUMBER OF RK06 CYLINDERS
         69F6 8F     3C  074D           MOVZWL  #411*3*22,-
         00B0 C5         0751            UCB$L_MAXBLOCK(R5)     ;SET MAXIMUM BLOCK NUMBER
     2364B006 8F     D0  0754           MOVL    #^X2364B006,-
         008C C5         075A            UCB$L_MEDIA_ID(R5)     ;SET MEDIA IDENT "DM RK06"
         00CC C5     B4  075D           CLRW    UCB$W_DM_DTYP(R5)       ;SET RK06 CONTROLLER DRIVE TYPE
 0A A4   0100 8F     B3  0761           BITW    #RK_DS_M_DDT,RK_DS(R4)  ;RK06 DRIVE?
              1C     13  0767           BEQL    10$                     ;IF EQL YES
              02     90  0769           MOVB    S^#DT$_RK07,-
           41 A5         076B            UCB$B_DEVTYPE(R5)      ;SET RK07 DEVICE TYPE
 46 A5   032F 8F     B0  076D           MOVW    #815,UCB$W_CYLINDERS(R5);SET NUMBER OF RK07 CYLINDERS
         D21E 8F     3C  0773           MOVZWL  #815*3*22,-
         00B0 C5         0777            UCB$L_MAXBLOCK(R5)     ;SET MAXIMUM BLOCK NUMBER
         008C C5     D6  077A           INCL    UCB$L_MEDIA_ID(R5)      ;SET MEDIA IDENT "DM RK07"
         0400 8F     A8  077E           BISW    #RK_CS1_M_CDT,-
         00CC C5         0782            UCB$W_DM_DTYP(R5)      ;SET RK07 CONTROLLER DRIVE TYPE
                     05  0785   10$:    RSB
```

```
                              0786  1413          .SBTTL  RK611-RK06/RK07 REGISTER DUMP ROUTINE
                              0786  1414  ;
                              0786  1415  ; DM_REGDUMP - RK611-RK06/RK07 REGISTER DUMP ROUTINE
                              0786  1416  ;
                              0786  1417  ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                              0786  1418  ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
                              0786  1419  ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                              0786  1420  ;
                              0786  1421  ; INPUTS:
                              0786  1422  ;
                              0786  1423  ;       R0 = ADDRESS OF REGISTER SAVE BUFFER.
                              0786  1424  ;       R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
                              0786  1425  ;       R5 = DEVICE UNIT UCB ADDRESS.
                              0786  1426  ;
                              0786  1427  ; OUTPUTS:
                              0786  1428  ;
                              0786  1429  ;       THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                              0786  1430  ;
                              0786  1431
                              0786  1432 DM_REGDUMP:                                 ;RK611-RK06/RK07 REGISTER DUMP ROUTINE
          80    12     D0     0786  1433          MOVL    #<RK_MR3+2-4+8>/2,(R0)+  ;INSERT NUMBER OF DEVICE REGISTERS
   51     00CE  C5     DE     0789  1434          MOVAL   UCB$W_DM_CS1(R5),R1      ;GET ADDRESS OF SAVED DEVICE REGISTERS
          52    0A     D0     078E  1435          MOVL    #<RK_MR1+2-4>/2,R2       ;SET NUMBER OF REGISTERS TO MOVE
          80    81     3C     0791  1436 10$:     MOVZWL  (R1)+,(R0)+              ;MOVE REGISTER TO BUFFER
             FA 52     F5     0794  1437          SOBGTR  R2,10$                   ;ANY MORE TO MOVE?
   80     00C4  C5     3C     0797  1438          MOVZWL  UCB$W_EC1(R5),(R0)+      ;INSERT ECC POSITION REGISTER
   80     00C6  C5     3C     079C  1439          MOVZWL  UCB$W_EC2(R5),(R0)+      ;INSERT ECC PATTERN REGISTER
          80    81     3C     07A1  1440          MOVZWL  (R1)+,(R0)+              ;INSERT MAINTENANCE REGISTER 2
          80    81     3C     07A4  1441          MOVZWL  (R1)+,(R0)+              ;INSERT MAINTENANCE REGISTER 3
          80    81     3C     07A7  1442          MOVZWL  (R1)+,(R0)+              ;INSERT DATAPATH NUMBER
          80    81     D0     07AA  1443          MOVL    (R1)+,(R0)+              ;INSERT DATAPATH REGISTER
          80    81     D0     07AD  1444          MOVL    (R1)+,(R0)+              ;INSERT FINAL MAP REGISTER
          80    61     D0     07B0  1445          MOVL    (R1),(R0)+               ;INSERT PREVIOUS MAP REGISTER
                05            07B3  1446          RSB                              ;
```

```
                           07B4          .SBTTL  RK06/RK07 DISK DRIVE INITIALIZATION
                           07B4   ;
                           07B4   ; DM_RKOX_INIT - RK06/RK07 DISK DRIVE INITIALIZATION
                           07B4   ;
                           07B4   ; THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
                           07B4   ; DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
                           07B4   ;
                           07B4   ; INPUTS:
                           07B4   ;
                           07B4   ;       R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
                           07B4   ;       R5 = DEVICE UNIT UCB ADDRESS.
                           07B4   ;
                           07B4   ; OUTPUTS:
                           07B4   ;
                           07B4   ;       UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
                           07B4   ;
                           07B4
                           07B4   DM_RKOX_INIT:                                     ;RK06/RK07 UNIT INITIALIZATION
      64   8000 8F    B0   07B4           MOVW    #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
   08 A4     54 A5    B0   07B9           MOVW    UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
      64        05    B0   07BE           MOVW    #F_DRVCLR!1,RK_CST(R4)    ;CLEAR DRIVE AND OBTAIN STATUS
              0098    30   07C1           BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
              FF7C    30   07C4           BSBW    DM_DTYPE                  ;CLASSIFY DRIVE TYPE
        53   64 A5    3C   07C7           MOVZWL  UCB$W_STS(R5),R3          ;SAVE CURRENT UNIT STATUS
   64 A5   0810 8F    AA   07CB           BICW    #UCB$M_ONLINE!UCB$M_VALID,UCB$W_STS(R5) ;SET UNIT OFFLINE/INVALID
   08 A4   1000 8F    B3   07D1           BITW    #RK_CS2_M_NED,RK_CS2(R4)  ;NONEXISTENT DISK?
                5E    12   07D7           BNEQ    50$                       ;IF NEQ YES
      64 A5     10    A8   07D9           BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
      43 53     08    E1   07DD           BBC     #UCB$V_VALID,R3,40$       ;IF CLR, VOLUME SOFTWARE INVALID
      64   8000 8F    B0   07E1   10$:    MOVW    #RK_CST_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
   08 A4     54 A5    B0   07E6           MOVW    UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
64    05   00CC C5    A9   07EB           BISW3   UCB$W_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE
              0068    30   07F1           BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
      64   8000 8F    B3   07F4           BITW    #RK_CS1_M_CERR,RK_CS1(R4) ;ANY CONTROLLER ERRORS?
                08    12   07F9           BNEQ    20$                       ;IF NEQ YES
   0A A4   0080 8F    B3   07FB           BITW    #RK_DS_M_DRDY,RK_DS(R4)   ;DRIVE READY?
                0B    12   0801           BNEQ    30$                       ;IF NEQ YES
       00000000'GF    16   0803   20$:    JSB     G^EXE$PWRTIMCHK           ;CHECK FOR MAXIMUM TIME EXCEEDED
             D5 50    E8   0809           BLBS    R0,10$                    ;IF LBS MORE TIME TO GO
                16    11   080C           BRB     40$                       ;
64    03   00CC C5    A9   080E   30$:    BISW3   UCB$W_DM_DTYP(R5),#F_PACKACK!1,RK_CS1(R4) ;ACKNOWLEDGE PACK
              0045    30   0814           BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
      64   8000 8F    B3   0817           BITW    #RK_CS1_M_CERR,RK_CS1(R4) ;ANY CONTROLLER ERRORS?
                06    12   081C           BNEQ    40$                       ;IF NEQ YES
   64 A5   0800 8F    A8   081E           BISW    #UCB$M_VALID,UCB$W_STS(R5) ;SET VOLUME SOFTWARE VALID
      64   8000 8F    B0   0824   40$:    MOVW    #RK_CST_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
   08 A4     54 A5    B0   0829           MOVW    UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
64    05   00CC C5    A9   082E           BISW3   UCB$W_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE
              0025    30   0834           BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
      64   8000 8F    B0   0837   50$:    MOVW    #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
        64     40 8F  9B   083C           MOVZBW  #RK_CS1_M_IE,RK_CS1(R4)   ;ENABLE DEVICE INTERRUPTS
                05         0840           RSB                               ;
```

```
                            0841         .SBTTL  RK611-RK06/RK07 UNSOLICITED INTERRUPT ROUTINE
                            0841  ;
                            0841  ; DM_UNSOLNT - RK611-RK06/RK07 UNSOLICITED INTERRUPT ROUTINE
                            0841  ;
                            0841  ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED FOR
                            0841  ; AN RK06 OR RK07 DRIVE.
                            0841  ;
                            0841  ; INPUTS:
                            0841  ;
                            0841  ;        R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
                            0841  ;        R5 = DEVICE UNIT UCB ADDRESS.
                            0841  ;
                            0841  ; OUTPUTS:
                            0841  ;
                            0841  ;        IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
                            0841  ;        UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
                            0841  ;        CLASSIFIED.
                            0841  ;
                            0841
                            0841  DM_UNSOLNT:                                      ;RK611-RK06/RK07 UNSOLICITED INTERRUPT
     64 A5   10       A8    0841         BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
            FEFB      30    0845         BSBW    DM_DTYPE                    ;CLASSIFY DRIVE TYPE
  0E 64 A5   0B       E1    0848         BBC     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF CLR, VOLUME SOFTWARE INVALID
0A A4   0080 8F       B3    084D         BITW    #RK_DS_M_DRDY,RK_DS(R4) ; DRIVE READY BIT SET?
              06      12    0853         BNEQ    10$                         ;IF NEQ YES
64 A5   0800 8F       AA    0855         BICW    #UCB$M_VALID,UCB$W_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
                      05    085B  10$:   RSB                                 ;
```

```
             085C              .SBTTL  WAIT FOR CONTROLLER READY
             085C       ;
             085C       ; DM_WAIT - WAIT FOR CONTROLLER READY
             085C       ;
             085C       ; THIS SUBROUTINE IS CALLED TO POLL THE RK611 CONTROLLER FOR READY. A MAX-
             085C       ; IMUM OF APPROXIMATELY 20 US ELAPSES BEFORE CONTROL IS RETURNED TO THE
             085C       ; CALLER.
             085C       ;
             085C
             085C  DM_WAIT:                                                ;WAIT FOR CONTROLLER READY
7E   50   7D 085C              MOVQ    R0,-(SP)                        ;SAVE R0, R1
             085F              DSBINT                                  ;DISABLE INTERRUPTS
             0865              TIMEWAIT        #2,#RK_CS1_M_RDY,RK_CS1(R4),W
             088A              ENBINT                                  ;ENABLE INTERRUPTS
50   8E   7D 088D              MOVQ    (SP)+,R0                        ;RESTORE R0, R1
          05 0890              RSB                                     ;
```

```
                              0891          .SBTTL  RK611 DISK CONTROLLER INTERRUPT DISPATCHER
                              0891  ;+
                              0891  ; DM$INT - RK611 DISK CONTROLLER INTERRUPT DISPATCHER
                              0891  ;
                              0891  ; THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
                              0891  ; ON AN RK611 DISK CONTROLLER. THE STATE OF THE STACK ON ENTRY IS:
                              0891  ;
                              0891  ;       00(SP) = ADDRESS OF IDB ADDRESS.
                              0891  ;       04(SP) = SAVED R2.
                              0891  ;       08(SP) = SAVED R3.
                              0891  ;       12(SP) = SAVED R4.
                              0891  ;       16(SP) = SAVED R5.
                              0891  ;       20(SP) = INTERRUPT PC.
                              0891  ;       24(SP) = INTERRUPT PSL.
                              0891  ;
                              0891  ; INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
                              0891  ;
                              0891  ;       IF THE INTERRUPTING CONTROLLER IS CURRENTLY OWNED AND THE OWNER
                              0891  ;       UNIT IS EXPECTING AN INTERRUPT, THEN THAT UNIT IS DISPATCHED FIRST.
                              0891  ;       ALL OTHER UNITS ARE DISPATCHED BY READING THE ATTENTION SUMMARY
                              0891  ;       REGISTER AND SCANNING FOR UNITS THAT HAVE ATTENTION SET. AS EACH
                              0891  ;       UNIT IS FOUND,  A TEST IS MADE TO DETERMINE IF AN INTERRUPT IS
                              0891  ;       EXPECTED ON THE UNIT. IF YES, THEN THE DRIVER IS CALLED AT ITS
                              0891  ;       INTERRUPT RETURN ADDRESS. ELSE THE DRIVER IS CALLED AT ITS UNSOL-
                              0891  ;       ICITED INTERRUPT ADDRESS. AS EACH CALL TO THE DRIVER RETURNS, THE
                              0891  ;       ATTENTION SUMMARY REGISTER IS REREAD AND AN ATTEMPT IS MADE TO FIND
                              0891  ;       ANOTHER UNIT TO DISPATCH. WHEN NO UNITS REQUESTING ATTENTION REMAIN,
                              0891  ;       THE INTERRUPT IS DISMISSED.
                              0891  ;-
                              0891
                              0891  DM$INT::                                        ;RK611 DISK CONTROLLER INTERRUPT DISPATCHER
         53   00 BE     D0    0891          MOVL    @(SP),R3                        ;GET ADDRESS OF IDB
            54    63    D0    0895          MOVL    IDB$L_CSR(R3),R4                ;GET ADDRESS OF CONTROL STATUS REGISTER 1
         55   04 A3     D0    0898          MOVL    IDB$L_OWNER(R3),R5              ;GET OWNER UNIT UCB ADDRESS
                  05    13    089C          BEQL    10$                             ;IF EQL NO OWNER
      3D 64 A5    01    E4    089E          BBSC    #UCB$V_INT,UCB$W_STS(R5),30$ ;IF SET, INTERRUPT EXPECTED
       64   8000 8F     B0    08A3  10$:    MOVW    #RK_CS1_M_CERR,RK_CS1(R4)  ;CLEAR CONTROLLER
         52   0E A4     3C    08A8          MOVZWL  RK_AS(R4),R2                    ;READ ATTENTION SUMMARY REGISTER
  52     52   08  08    EA    08AC          FFS     #8,#8,R2,R2                     ;FIND FIRST UNIT REQUESTING ATTENTION
                  11    12    08B1          BNEQ    20$                             ;IF NEQ UNIT FOUND
         64   40 8F     9B    08B3          MOVZBW  #RK_CS1_M_IE,RK_CS1(R4)         ;ENABLE DEVICE INTERRUPTS
            5E    04    C0    08B7          ADDL    #4,SP                           ;CLEAN STACK
            50    8E    7D    08BA          MOVQ    (SP)+,R0                        ;RESTORE REGISTERS
            52    8E    7D    08BD          MOVQ    (SP)+,R2                        ;
            54    8E    7D    08C0          MOVQ    (SP)+,R4                        ;
                  02          08C3          REI                                     ;
            52    08    C2    08C4  20$:    SUBL    #8,R2                           ;CALCULATE UNIT NUMBER
         08 A4    52    B0    08C7          MOVW    R2,RK_CS2(R4)                   ;SET UNIT NUMBER
         55   18 A342   D0    08CB          MOVL    IDB$L_UCBLST(R3)[R2],R5         ;GET ADDRESS OF UCB
                  79    13    08D0          BEQL    80$                             ;IF EQL NO CORRESPONDING UNIT
  64     01   00CC C5   A9    08D2          BISW3   UCB$W_DM_DTYP(R5),#F_NOP!1,RK_CS1(R4) ;SELECT DRIVE AND GET STATUS
                FF81    30    08D8          BSBW    DM_WAIT                         ;WAIT FOR CONTROLLER READY
      6D 64 A5    01    E5    08DB          BBCC    #UCB$V_INT,UCB$W_STS(R5),90$ ;IF CLR, INTERRUPT NOT EXPECTED
       0093 C5    0E    91    08E0  30$:    CMPB    #CDF_READHEAD,UCB$B_CEX(R5) ;READ HEADER FUNCTION?
                  12    12    08E5          BNEQ    40$                             ;IF NEQ NO
    00F4 C5    14 A4    B0    08E7          MOVW    RK_DB(R4),UCB$W_DM_DB(R5) ;SAVE SECTOR HEADER INFORMATION
    00F6 C5    14 A4    B0    08ED          MOVW    RK_DB(R4),UCB$W_DM_DB+2(R5) ;
```

```
        00F8 C5    14 A4   B0  08F3   1603         MOVW    RK_DB(R4),UCB$W_DM_DB+4(R5) ;
              52    64   9E  08F9   1604 40$:    MOVAB   RK_CS1(R4),R2             ;GET ADDRESS OF CONTROL STATUS REGISTER 1
        53    00CE  C5   9E  08FC   1605         MOVAB   UCB$W_DM_CS1(R5),R3       ;GET ADDRESS OF REGISTER SAVE AREA
              83    82   B0  0901   1606         MOVW    (R2)+,(R3)+               ;SAVE CONTROL STATUS REGISTER 1
                    7D   19  0904   1607         BLSS    120$                      ;IF LSS ERROR ENCOUNTERED
     78 68 A5       01   E0  0906   1608         BBS     #UCB$V_DIAGBUF,UCB$W_DEVSTS(R5),120$ ;IF SET, DIAGNOSTIC BUFFER
                    08   91  090B   1609         CMPB    #CDF_PACKACK,-            ; PACK ACKNOWLEDGE FUNCTION?
              0093  C5       090D   1610                 UCB$B_CEX(R5)             ;
                    0F   12  0910   1611         BNEQ    50$                       ; BRANCH IF NOT.
     0A A4    0080  8F   B3  0912   1612         BITW    #RK_DS_M_DRDY,RK_DS(R4)   ; DRIVE READY BIT SET?
                    69   13  0918   1613         BEQL    120$                      ; ERROR IF NOT.
              0080  8F   A8  091A   1614         BISW    #RK_DS_M_DRDY,-           ; SAVE READY BIT IN UCB.
              00D8  C5       091E   1615                 UCB$W_DM_DS(R5)           ;
              83    82   B0  0921   1616 50$:    MOVW    (R2)+,(R3)+               ;SAVE WORD COUNT REGISTER
              63    62   B0  0924   1617         MOVW    (R2),(R3)                 ;SAVE BUFFER ADDRESS REGISTER
        53    10    A5   7D  0927   1618 60$:    MOVQ    UCB$L_FR3(R5),R3          ;RESTORE DRIVER CONTEXT
              0C    B5   16  092B   1619         JSB     @UCB$L_FPC(R5)            ;CALL DRIVER AT INTERRUPT RETURN ADDRESS
        53    00    BE   D0  092E   1620         MOVL    @(SP),R3                  ;GET ADDRESS OF IDB
              54    63   D0  0932   1621         MOVL    IDB$L_CSR(R3),R4          ;GET ADDRESS OF CONTROL STATUS REGISTER 1
        64    8000  8F   B0  0935   1622 70$:    MOVW    #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER
        08 A4   54  A5   B0  093A   1623         MOVW    UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
  64    05    00CC  C5   A9  093F   1624         BISW3   UCB$W_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE ERRORS
              FF14       30  0945   1625         BSBW    DM_WAIT                   ;WAIT FOR CONTROLLER READY
              FF58       31  0948   1626         BRW     10$                       ;
                    18   11  094B   1627 80$:    BRB     100$                      ;
              FEF1       30  094D   1628 90$:    BSBW    DM_UNSOLNT                ;CALL UNSOLICITED INTERRUPT ROUTINE
        53    00    BE   D0  0950   1629         MOVL    @(SP),R3                  ;GET ADDRESS OF IDB
              54    63   D0  0954   1630         MOVL    IDB$L_CSR(R3),R4          ;GET ADDRESS OF CONTROL STATUS REGISTER 1
        64    8000  8F   B3  0957   1631         BITW    #RK_CS1_M_CERR,RK_CS1(R4) ;ANY ERROR CONDITION PRESENT?
                    D7   13  095C   1632         BEQL    70$                       ;IF EQL NO
        08 A4       20   B0  095E   1633         MOVW    #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR ENTIRE RK611 SUBSYSTEM
              FF3E       31  0962   1634         BRW     10$                       ;
        64    0405  8F   B0  0965   1635 100$:   MOVW    #RK_CS1_M_CDT!F_DRVCLR!1,RK_CS1(R4) ;CLEAR RK07 DRIVE
              FEEF       30  096A   1636         BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
                    64   B5  096D   1637         TSTW    RK_CS1(R4)                ;SUCCESSFUL COMPLETION?
                    0F   18  096F   1638         BGEQ    110$                      ;IF GEQ YES
        64    8000  8F   B0  0971   1639         MOVW    #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER
        08 A4       52   B0  0976   1640         MOVW    R2,RK_CS2(R4)             ;SET UNIT NUMBER
           64       05   B0  097A   1641         MOVW    #F_DRVCLR!1,RK_CS1(R4)    ;CLEAR RK06 DRIVE
              FEDC       30  097D   1642         BSBW    DM_WAIT                   ;WAIT FOR FUNCTION TO COMPLETE
              FF20       31  0980   1643 110$:   BRW     10$                       ;
              83    82   B0  0983   1644 120$:   MOVW    (R2)+,(R3)+               ;SAVE WORD COUNT REGISTER
              83    82   B0  0986   1645         MOVW    (R2)+,(R3)+               ;SAVE BUFFER ADDRESS REGISTER
              83    82   B0  0989   1646         MOVW    (R2)+,(R3)+               ;SAVE DESIRED SECTOR/TRACK ADDRESS REGISTER
              83    82   B0  098C   1647         MOVW    (R2)+,(R3)+               ;SAVE CONTROL STATUS REGISTER 2
              83    82   B0  098F   1648         MOVW    (R2)+,(R3)+               ;SAVE DRIVE STATUS REGISTER
              83    82   B0  0992   1649         MOVW    (R2)+,(R3)+               ;SAVE ERROR REGISTER
              83    82   B0  0995   1650         MOVW    (R2)+,(R3)+               ;SAVE ATTENTION SUMMARY/OFFSET REGISTER
              83    82   B0  0998   1651         MOVW    (R2)+,(R3)+               ;SAVE DESIRED CYLINDER ADDRESS REGISTER
              52    04   C0  099B   1652         ADDL    #4,R2                     ;POINT TO MAINTENANCE REGISTER 1
              83    82   B0  099E   1653         MOVW    (R2)+,(R3)+               ;SAVE MAINTENANCE REGISTER 1
        00C4  C5    82   B0  09A1   1654         MOVW    (R2)+,UCB$W_EC1(R5)       ;SAVE ECC POSITION REGISTER
        00C6  C5    82   B0  09A6   1655         MOVW    (R2)+,UCB$W_EC2(R5)       ;SAVE ECC PATTERN REGISTER
              83    82   B0  09AB   1656         MOVW    (R2)+,(R3)+               ;SAVE MAINTENANCE REGISTER 2
              63    62   B0  09AE   1657         MOVW    (R2),(R3)                 ;SAVE MAINTENANCE REGISTER 3
     04 00D6  C5    09   E1  09B1   1658         BBC     #RK_CS2_V_MDS,UCB$W_DM_CS2(R5),130$ ;IF CLR, NO MULTI-DRIVE SELECT
        08 A4       20   B0  09B7   1659         MOVW    #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR ENTIRE SUBSYSTEM
```

```
                      FF69   31  0988  1660 130$:    BRW     60$                       ;
                      09BE          .SBTTL  RK611 DISK CONTROLLER INITIALIZATION
                      09BE  ;+
                      09BE  ; DM_RK611_INIT - RK611 DISK CONTROLLER INITIALIZATION
                      09BE  ;
                      09BE  ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                      09BE  ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF RK611 DISK CONTROLLERS.
                      09BE  ;
                      09BE  ; INPUTS:
                      09BE  ;
                      09BE  ;       R0 = SCRATCH.
                      09BE  ;       R1 = SCRATCH.
                      09BE  ;       R2 = SCRATCH.
                      09BE  ;       R3 = SCRATCH.
                      09BE  ;       R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
                      09BE  ;       R5 = ADDRESS OF CONTROLLER IDB.
                      09BE  ;
                      09BE  ;       ALL INTERRUPTS ARE LOCKED OUT.
                      09BE  ;
                      09BE  ; OUTPUTS:
                      09BE  ;
                      09BE  ;       THE RK611 CONTROLLER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                      09BE  ;-
                      09BE
                      09BE  DM_RK611_INIT:                          ;RK611 DISK CONTROLLER INITIALIZATION
08 A4    20     B0    09BE          MOVW    #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR CONTROLLER AND ALL DRIVES
64    40 8F     9B    09C2          MOVZBW  #RK_CS1_M_IE,RK_CS1(R4) ;ENABLE DEVICE INTERRUPTS
                05    09C6          RSB                             ;
```

```
                              09C7         .SBTTL  RK611 Autoconfigure Unit Delivery Routine
                              09C7    ;+
                              09C7    ; DM$DELIVER - RK611 Autoconfigure Unit Delivery Routine
                              09C7    ;
                              09C7    ; This routine is called by the SYSGEN AUTOCONFIGURE command to determine
                              09C7    ; which RK611 unit numbers to configure.  It is called once for each possible
                              09C7    ; unit, 0 through 7.
                              09C7    ;
                              09C7    ; INPUTS:
                              09C7    ;       R3 Address of controller IDB, or zero if none exists
                              09C7    ;       R4 Address of CSR
                              09C7    ;       R5 Unit number which this routine must decide whether or not to
                              09C7    ;          configure
                              09C7    ;       R6 Base addres of UNIBUS adapter I/O space
                              09C7    ;       R7 Address of AUTOCONFIGURE ACF
                              09C7    ;       R8 Address of UNIBUS ADP
                              09C7    ;       IPL = 31
                              09C7    ;
                              09C7    ; OUTPUTS:
                              09C7    ;       R0 TRUE  ==> configure unit indicated in R5
                              09C7    ;          FALSE ==> do not configure unit indicated in R5
                              09C7    ;
                              09C7    ; Interference with "normal" data transfers is a major concern for this unit
                              09C7    ; delivery routine.  Since it is called without the controls of the QIO
                              09C7    ; mechanism, the state of data transfers when it is entered is unpredictable.
                              09C7    ; Experience has shown that conditions are so unpredictable that the only
                              09C7    ; option open to this routine is forcing all current activitiy to be retried.
                              09C7    ; To this end, all UCBs listed in the IDB passed to this routine, if any, are
                              09C7    ; made to appear as if a power failure has occured.  Having done this, the
                              09C7    ; retrying of currently active opeations is relative assured.  We do not
                              09C7    ; simulate a power failure to the extent of calling the controller and unit
                              09C7    ; initialization routines; after all, this routine determines the state of the
                              09C7    ; controller and its units quite completely.
                              09C7    ;
                              09C7    ;-
                              09C7
                              09C7   DM$DELIVER:
                     55   D5  09C7          TSTL    R5                          ; Is this the first call for this
                     02   12  09C9          BNEQ    10$                         ; for this controller?  If so, get
                     07   10  09CB          BSBB    GET_UNITS                   ; complete units present information.
50   24 A7   01      55   EF  09CD   10$:   EXTZV   R5, #1, -                   ; For each unit, get presense data from
                              09D3                  ACF$L_DLVR_SCRH(R7), R0     ; information prepared by GET_UNITS.
                          05  09D3          RSB                                 ; Then, return to AUTOCONFIGURE.
                              09D4
                              09D4    ;+
                              09D4    ; NB: the use of the one-time get-units-information routine GET_UNITS reduces
                              09D4    ; to one the number of times we must fool with the controller and thus our
                              09D4    ; pontential for munging "normal" opeations.  It also insures that all
                              09D4    ; controller munging occurs at the same time that a power failure is
                              09D4    ; simulated.
                              09D4    ;-
                              09D4
                              09D4   GET_UNITS:
                              09D4          DSBINT                              ; Insure no interruptions.
                              09DA
                              09DA    ;+
                              09DA    ; SIMULATE A POWER FAILURE ON ALL KNOWN UCBs
                          09DA  ;-
                53    D5  09DA               TSTL    R3                          ; Is there an IDB?  If not, the RK611
                23    13  09DC               BEQL    500$                        ; must be inactive.
       50    0C A3    3C  09DE               MOVZWL  IDB$W_UNITS(R3), R0         ; Get count of UCBs to test.
    51    14 A340     D0  09E2  10$:         MOVL    IDB$L_UCBLST-4(R3)[R0], R1 ; Get a UCB address.
                15    13  09E7               BEQL    19$                         ; Branch if not really a UCB address.
       64 A1    20    A8  09E9               BISW    #UCB$M_POWER, UCB$W_STS(R1) ; Pretend power failure.
       64 A1    03    B3  09ED               BITW    #<UCB$M_INT!UCB$M_TIM>, -   ; Test for timeout in progress.
                          09F1                       UCB$W_STS(R1)
                0B    13  09F1               BEQL    19$                         ; Branch if no timeout in progress.
       64 A1    02    AA  09F3               BICW    #UCB$M_INT, UCB$W_STS(R1)  ; Clear interrupt expected.
       64 A1    01    A8  09F7               BISW    #UCB$M_TIM, UCB$W_STS(R1)  ; Indicate that a timeout is expected
             6C A1    D4  09FB               CLRL    UCB$L_DUETIM(R1)            ; immediately.
             E1 50    F5  09FE  19$:         SOBGTR  R0, 10$                     ; Loop through all UCBs.
                          0A01
                          0A01  ;+
                          0A01  ; DISCOVERING WHICH UNITS ARE PRESENT
                          0A01  ;-
             24 A7    D4  0A01  500$:        CLRL    ACF$L_DLVR_SCRH(R7)         ; Clear all units present bits.
                50    D4  0A04               CLRL    R0                          ; Initialize unit number.
              FE53    30  0A06  600$:        BSBW    DM_WAIT                     ; Wait for controller ready.
    64     8000 8F    B0  0A09               MOVW    #RK_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
       08 A4    50    B0  0A0E               MOVW    R0, RK_CS2(R4)              ; Set unit number.
          64    01    B0  0A12               MOVW    #1, RK_CS1(R4)              ; Select drive and get status.
              FE44    30  0A15               BSBW    DM_WAIT                     ; Wait for controller ready.
 08 A4     1000 8F    B3  0A18               BITW    #RK_CS2_M_NED, RK_CS2(R4)  ; Nonexistent drive?
                26    12  0A1E               BNEQ    690$                        ; If nonexistent, no more to do here.
   00 24 A7     50    E2  0A20               BBSS    R0, ACF$L_DLVR_SCRH(R7), 610$ ; Set device present bit.
    64     8000 8F    B0  0A25  610$:        MOVW    #RK_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
       08 A4    50    B0  0A2A               MOVW    R0, RK_CS2(R4)              ; Set unit number.
    64     0405 8F    B0  0A2E               MOVW    #^X405, RK_CS1(R4)          ; Clear drive as a RK07.
              FE26    30  0A33               BSBW    DM_WAIT                     ; Wait for function to complete.
                64    B5  0A36               TSTW    RK_CS1(R4)                  ; Controller errors?  Errors mean its a
                0C    18  0A38               BGEQ    690$                        ; RK06 and must be cleared differently.
    64     8000 8F    B0  0A3A               MOVW    #RK_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
       08 A4    50    B0  0A3F               MOVW    R0, RK_CS2(R4)              ; Set unit number.
          64    05    B0  0A43               MOVW    #5, RK_CS1(R4)              ; Clear drive as a RK06.
       BC 50    07    F3  0A46  690$:        AOBLEQ  #7, R0, 600$                ; Loop over all possible drives.
              FE0F    30  0A4A               BSBW    DM_WAIT                     ; Wait for last operation to complete.
    64     8000 8F    B0  0A4D               MOVW    #RK_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
                          0A52               ENBINT                              ; Restore previous interrupt state.
                      05  0A55               RSB                                 ; Return to main unit-deliver routine.
                          0A56
                          0A56  DM_END:                                         ;ADDRESS OF LAST LOCATION IN DRIVER
                          0A56
                          0A56               .END
```

```
$$$                            = 00000020 R     02    DM_M_ECI                       = 00000001
$$OP                           = 00000002             DM_REGDUMP                       00000786 R     03
ACF$L_DLVR_SCRH                = 00000024             DM_RK0X_INIT                     000007B4 R     03
ACP$ACCESS                       ********     X 03    DM_RK61T_INIT                    000009BE R     03
ACP$DEACCESS                     ********     X 03    DM_STARTIO                       00000111 R     03
ACP$MODIFY                       ********     X 03    DM_UNSOLNT                       00000841 R     03
ACP$MOUNT                        ********     X 03    DM_V_DCK                       = 00000001
ACP$READBLK                      ********     X 03    DM_V_ECC_DEFER                 = 00000002
ACP$WRITEBLK                     ********     X 03    DM_V_ECI                       = 00000000
APPLY_ECC                        000002A7 R     03    DM_WAIT                          0000085C R     03
AT$_UBA                        = 00000001             DPT$C_LENGTH                   = 00000038
AVAILABLE                        000001C8 R     03    DPT$C_VERSION                  = 00000004
CDF_AVAILABLE                  = 0000000F             DPT$INITAB                       00000038 R     02
CDF_DRVCLR                     = 00000004             DPT$M_SVP                      = 00000002
CDF_NOP                        = 00000000             DPT$REINITAB                     00000072 R     02
CDF_OFFSET                     = 00000006             DPT$TAB                          00000000 R     02
CDF_PACKACK                    = 00000008             DRVCLR                           000001DE R     03
CDF_READDATA                   = 0000000C             DT$_RK06                       = 00000001
CDF_READHEAD                   = 0000000E             DT$_RK07                       = 00000002
CDF_RECAL                      = 00000003             DYN$C_CRB                      = 00000005
CDF_RELEASE                    = 00000005             DYN$C_DDB                      = 00000006
CDF_RETCENTER                  = 00000007             DYN$C_DPT                      = 0000001E
CDF_SEEK                       = 00000002             DYN$C_UCB                      = 00000010
CDF_STARTSPNDL                 = 00000009             ECC                              0000026A R     03
CDF_UNLOAD                     = 00000001             EMB$L_DV_REGSAV                = 0000004E
CDF_WRITECHECK                 = 0000000A             ENBXIT                           000004DC R     03
CDF_WRITEDATA                  = 0000000B             ERL$DEVICERR                     ********     X 03
CDF_WRITEHEAD                  = 0000000D             ERL$DEVICTMO                     ********     X 03
CHECKRETRY                       00000237 R     03    EXE$ABORTIO                      ********     X 03
CHECKXT                          00000241 R     03    EXE$GL_TENUSEC                   ********     X 03
CRB$L_INTD                     = 00000024             EXE$GL_UBDELAY                   ********     X 03
DATACHECK                        00000209 R     03    EXE$IOFORK                       ********     X 03
DC$_DISK                       = 00000001             EXE$LCLDSKVALID                  ********     X 03
DDB$K_CART                     = 00000002             EXE$ONEPARM                      ********     X 03
DDB$L_ACPD                     = 00000010             EXE$PWRTIMCHK                    ********     X 03
DDB$L_DDT                      = 0000000C             EXE$SENSEMODE                    ********     X 03
DEFER_ECC                        000002D2 R     03    EXE$SETCHAR                      ********     X 03
DEV$M_AVL                      = 00040000             EXE$ZEROPARM                     ********     X 03
DEV$M_DIR                      = 00000008             EXEC_FUNCTION                    000001DE R     03
DEV$M_ELG                      = 00400000             FATALERR                         0000034B R     03
DEV$M_FOD                      = 00004000             FDISPATCH                        0000017E R     03
DEV$M_IDV                      = 04000000             FEXH                             00000406 R     03
DEV$M_NNM                      = 00000200             FEXL                             0000040F R     03
DEV$M_ODV                      = 08000000             FTAB                             00000038 R     03
DEV$M_RND                      = 10000000             FUNCTAB_LEN                    = 000000A0
DEV$M_SHR                      = 00010000             FUNCXT                           000003C1 R     03
DM$DDT                           00000000 RG    03    F_AVAILABLE                    = 00000000
DM$DELIVER                       000009C7 R     03    F_DRVCLR                       = 00000004
DM$INT                           00000891 RG    03    F_NOP                          = 00000000
DM_BYTECNT                       00000100 R     03    F_OFFSET                       = 0000000C
DM_DTYPE                         00000743 R     03    F_PACKACK                      = 00000002
DM_END                           00000A56 R     03    F_READDATA                     = 00000010
DM_FUNCTABLE                     00000060 R     03    F_READHEAD                     = 00000014
DM_IND_M_OF                    = 00000001             F_RECAL                        = 0000000A
DM_IND_V_OF                    = 00000000             F_RELEASE                      = 00000000
DM_M_DCK                       = 00000002             F_RETCENTER                    = 0000000C
DM_M_ECC_DEFER                 = 00000004             F_SEEK                         = 0000000E
```

```
F_STARTSPNDL                    = 00000008                  IOC$REQDATAP                      ********    X   03
F_UNLOAD                        = 00000006                  IOC$REQMAPREG                     ********    X   03
F_WRITECHECK                    = 00000018                  IOC$REQPCHANH                     ********    X   03
F_WRITEDATA                     = 00000012                  IOC$REQPCHANL                     ********    X   03
F_WRITEHEAD                     = 00000016                  IOC$RETURN                        ********    X   03
GET_UNITS                         000009D4 R      03        IOC$UPDATRANSP                    ********    X   03
IDB$L_CSR                       = 00000000                  IOC$WFIKPCH                       ********    X   03
IDB$L_OWNER                     = 00000004                  IOC$WFIRLCH                       ********    X   03
IDB$L_UCBLST                    = 00000018                  IRP$L_MEDIA                     = 00000038
IDB$W_UNITS                     = 0000000C                  IRP$L_SVAPTE                    = 0000002C
IMMED                             00000444 R      03        IRP$S_FCODE                     = 00000006
IO$M_DATACHECK                  = 00004000                  IRP$V_DIAGBUF                   = 00000007
IO$V_DATACHECK                  = 0000000E                  IRP$V_FCODE                     = 00000000
IO$V_INHRETRY                   = 0000000F                  IRP$V_PHYSIO                    = 00000008
IO$V_INHSEEK                    = 0000000C                  IRP$W_BCNT                      = 00000032
IO$_ACCESS                      = 00000032                  IRP$W_FUNC                      = 00000020
IO$_ACPCONTROL                  = 00000038                  IRP$W_STS                       = 0000002A
IO$_AVAILABLE                   = 00000011                  MASKH                           = 00000008
IO$_CREATE                      = 00000033                  MASKL                           = 04000000
IO$_DEACCESS                    = 00000034                  NOP                               000001DE R      03
IO$_DELETE                      = 00000035                  NORMAL                            0000023E R      03
IO$_DRVCLR                      = 00000004                  OFF                               000002D7 R      03
IO$_MODIFY                      = 00000036                  OFFSET                            000001DE R      03
IO$_MOUNT                       = 00000039                  OFFSETERR                         00000346 R      03
IO$_NOP                         = 00000000                  OFFSIZ                          = 00000008
IO$_OFFSET                      = 00000006                  OFFTAB                            00000058 R      03
IO$_PACKACK                     = 00000008                  PACKACK                           000001D8 R      03
IO$_READHEAD                    = 0000000E                  POSIT                             00000472 R      03
IO$_READLBLK                    = 00000021                  PR$_IPL                         = 00000012
IO$_READPBLK                    = 0000000C                  READDATA                          000001F3 R      03
IO$_READVBLK                    = 00000031                  READHEAD                          000001E7 R      03
IO$_RECAL                       = 00000003                  RECAL                             000001DE R      03
IO$_RELEASE                     = 00000005                  RELEASE                           000001DE R      03
IO$_RETCENTER                   = 00000007                  RELES                             00000441 R      03
IO$_SEEK                        = 00000002                  RESETXFR                          00000724 R      03
IO$_SENSECHAR                   = 0000001B                  RETCENTER                         000001DE R      03
IO$_SENSEMODE                   = 00000027                  RETREG                            00000619 R      03
IO$_SETCHAR                     = 0000001A                  RETRY                             00000267 R      03
IO$_SETMODE                     = 00000023                  RETRYERR                          0000032F R      03
IO$_STARTSPNDL                  = 00000019                  RK_AS                             0000000E
IO$_UNLOAD                      = 00000001                  RK_BA                             00000004
IO$_VIRTUAL                     = 0000003F                  RK_CS1                            00000000
IO$_WRITECHECK                  = 0000000A                  RK_CS1_M_CDT                    = 00000400
IO$_WRITEHEAD                   = 0000000D                  RK_CS1_M_CERR                   = 00008000
IO$_WRITELBLK                   = 00000020                  RK_CS1_M_CTO                    = 00000800
IO$_WRITEPBLK                   = 0000000B                  RK_CS1_M_DPPE                   = 00000020
IO$_WRITEVBLK                   = 00000030                  RK_CS1_M_GO                     = 00000001
IOC$APPLYECC                      ********    X   03        RK_CS1_M_IE                     = 00000040
IOC$DIAGBUFILL                    ********    X   03        RK_CS1_M_RDY                    = 00000080
IOC$LOADUBAMAP                    ********    X   03        RK_CS1_M_SPAR                   = 00002000
IOC$MNTVER                        ********    X   03        RK_CS1_V_CERR                   = 0000000F
IOC$MOVTOUSER                     ********    X   03        RK_CS2                            00000008
IOC$PURGDATAP                     ********    X   03        RK_CS2_M_DLT                    = 00008000
IOC$RELCHAN                       ********    X   03        RK_CS2_M_MDS                    = 00000200
IOC$RELDATAP                      ********    X   03        RK_CS2_M_NED                    = 00001000
IOC$RELMAPREG                     ********    X   03        RK_CS2_M_NEM                    = 00000800
IOC$REQCOM                        ********    X   03        RK_CS2_M_PGE                    = 00000400
```

```
RK_CS2_M_RLS                   = 00000008                    SS$_NORMAL                     = 00000001
RK_CS2_M_SCLR                  = 00000020                    SS$_PARITY                     = 000001F4
RK_CS2_M_UFE                   = 00000100                    SS$_TIMEOUT                    = 0000022C
RK_CS2_M_UPE                   = 00002000                    SS$_UNSAFE                     = 0000023C
RK_CS2_M_WCE                   = 00004000                    SS$_VOLINV                     = 00000254
RK_CS2_V_MDS                   = 00000009                    SS$_WASECC                     = 00000639
RK_CS2_V_NED                   = 0000000C                    SS$_WRITLCK                    = 0000025C
RK_CS2_V_UPE                   = 0000000D                    STARTSPNDL                       000001DE R      03
RK_CS2_V_WCE                   = 0000000E                    TRANSFR                          00000200 R      03
RK_DA                            00000006                    TRANXT                           00000244 R      03
RK_DB                            00000014                    UCB$B_CEX                      = 00000093
RK_DC                            00000010                    UCB$B_DEVCLASS                 = 00000040
RK_DS                            0000000A                    UCB$B_DEVTYPE                  = 00000041
RK_DS_M_DDT                    = 00000100                    UCB$B_DIPL                     = 0000005E
RK_DS_M_DRDY                   = 00000080                    UCB$B_DM_IND                     000000FA
RK_DS_M_DSC                    = 00004000                    UCB$B_ERTCNT                   = 00000080
RK_DS_M_VV                     = 00000040                    UCB$B_ERTMAX                   = 00000081
RK_DS_V_DRDY                   = 00000007                    UCB$B_FEX                      = 00000092
RK_DS_V_VV                     = 00000006                    UCB$B_FIPL                     = 0000000B
RK_EC1                           00000018                    UCB$B_OFFNDX                   = 000000CA
RK_EC2                           0000001A                    UCB$B_OFFRTC                   = 000000CB
RK_ER                            0000000C                    UCB$B_SECTORS                  = 00000044
RK_ER_M_BSE                    = 00000080                    UCB$B_TRACKS                   = 00000045
RK_ER_M_COE                    = 00000200                    UCB$K_DM_LENGTH                = 00000100
RK_ER_M_DCK                    = 00008000                    UCB$K_LCC_DISK_LENGTH          = 000000CC
RK_ER_M_DRPAR                  = 00000008                    UCB$L_CRB                      = 00000024
RK_ER_M_DTE                    = 00001000                    UCB$L_DEVCHAR                  = 00000038
RK_ER_M_DTYE                   = 00000020                    UCB$L_DEVCHAR2                 = 0000003C
RK_ER_M_ECH                    = 00000040                    UCB$L_DM_DPR                     000000E8
RK_ER_M_FMTE                   = 00000010                    UCB$L_DM_FMPR                    000000EC
RK_ER_M_HVRC                   = 00000100                    UCB$L_DM_FRS                     000000FB
RK_ER_M_IDAE                   = 00000400                    UCB$L_DM_PMPR                    000000F0
RK_ER_M_ILF                    = 00000001                    UCB$L_DPC                      = 0000009C
RK_ER_M_NXF                    = 00000004                    UCB$L_DUETIM                   = 0000006C
RK_ER_M_OPI                    = 00002000                    UCB$L_FPC                      = 0000000C
RK_ER_M_SKI                    = 00000002                    UCB$L_FR3                      = 00000010
RK_ER_M_WLE                    = 00000800                    UCB$L_IRP                      = 00000058
RK_ER_V_DRPAR                  = 00000003                    UCB$L_MAXBLOCK                 = 000000B0
RK_ER_V_HVRC                   = 00000008                    UCB$L_MEDIA_ID                 = 0000008C
RK_ER_V_UNS                    = 0000000E                    UCB$L_SVAPTE                   = 00000078
RK_ER_V_WLE                    = 0000000B                    UCB$M_DIAGBUF                  = 00000002
RK_MR1                           00000016                    UCB$M_ECC                      = 00000001
RK_MR2                           0000001C                    UCB$M_INT                      = 00000002
RK_MR3                           0000001E                    UCB$M_ONLINE                   = 00000010
RK_SPR                           00000012                    UCB$M_POWER                    = 00000020
RK_WC                            00000002                    UCB$M_TIM                      = 00000001
RLSCHN                           00000613 R      03          UCB$M_TIMOUT                   = 00000040
SEEK                             000001DE R      03          UCB$M_VALID                    = 00000800
SIZ...                         = 00000001                    UCB$V_DIAGBUF                  = 00000001
SS$_CTRLERR                    = 00000054                    UCB$V_ECC                      = 00000000
SS$_DATACHECK                  = 0000005C                    UCB$V_INT                      = 00000001
SS$_DRVERR                     = 0000008C                    UCB$V_POWER                    = 00000005
SS$_FORMAT                     = 000000BC                    UCB$V_VALID                    = 0000000B
SS$_IVADDR                     = 00000134                    UCB$W_BCNT                     = 0000007E
SS$_IVBUFLEN                   = 0000034C                    UCB$W_BCR                      = 000000C0
SS$_MEDOFL                     = 000001A4                    UCB$W_BOFF                     = 0000007C
SS$_NONEXDRV                   = 000001C4                    UCB$W_CYLINDERS                = 00000046
```

```
UCB$W_DA                 = 000000BC
UCB$W_DC                 = 000000BE
UCB$W_DEVBUFSIZ          = 00000042
UCB$W_DEVSTS             = 00000068
UCB$W_DM_AS                000000DC
UCB$W_DM_BA                000000D2
UCB$W_DM_CS1               000000CE
UCB$W_DM_CS2               000000D6
UCB$W_DM_DA                000000D4
UCB$W_DM_DB                000000F4
UCB$W_DM_DC                000000DE
UCB$W_DM_DPN               000000E6
UCB$W_DM_DS                000000D8
UCB$W_DM_DTYP              000000CC
UCB$W_DM_ER                000000DA
UCB$W_DM_MR1               000000E0
UCB$W_DM_MR2               000000E2
UCB$W_DM_MR3               000000E4
UCB$W_DM_WC                000000D0
UCB$W_ECT                = 000000C4
UCB$W_EC2                = 000000C6
UCB$W_FUNC               = 0000009A
UCB$W_OFFSET             = 000000C8
UCB$W_STS                = 00000064
UCB$W_UNIT               = 00000054
UNLOAD                     000001D0 R      03
VEC$B_DATAPATH           = 00000013
VEC$L_IDB                = 00000008
VEC$L_INITIAL            = 0000000C
VEC$L_UNITINIT           = 00000018
VEC$S_DATAPATH           = 00000005
VEC$S_MAPREG             = 0000000F
VEC$V_DATAPATH           = 00000000
VEC$V_MAPREG             = 00000000
VEC$W_MAPREG             = 00000010
WRITECHECK                 000001E7 R      03
WRITEDATA                  000001EE R      03
WRITEHEAD                  000001E7 R      03
XFER                       000004E2 R      03

                                        +-----------------+
                                        ! Psect synopsis !
                                        +-----------------+

PSECT name                 Allocation          PSECT No.  Attributes
----------                 ----------          ---------  ----------
.  ABS  .                  00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                      00000100  (   256.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$105_PROLOGUE            00000087  (   135.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$115_DRIVER              00000A56  (  2646.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
```

```
                                              +--------------------------+
                                              ! Performance indicators !
                                              +--------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        34    00:00:00.07    00:00:01.40
Command processing                   118    00:00:00.38    00:00:04.95
Pass 1                               629    00:00:20.47    00:01:18.24
Symbol table sort                      0    00:00:02.69    00:00:10.91
Pass 2                               326    00:00:04.56    00:00:18.34
Symbol table output                   47    00:00:00.26    00:00:01.50
Psect synopsis output                  2    00:00:00.02    00:00:00.08
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                1158    00:00:28.45    00:01:55.43

The working set limit was 2250 pages.
164077 bytes (321 pages) of virtual memory were used to buffer the intermediate code.
There were 130 pages of symbol table space allocated to hold 2395 non-local and 84 local symbols.
1792 source lines were read in Pass 1, producing 23 object records in Pass 2.
56 pages of virtual memory were used to define 53 macros.

                                              +----------------------------+
                                              ! Macro library statistics !
                                              +----------------------------+

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                37
_$255$DUA28:[SYSLIB]STARLET.MLB;2                             10
TOTALS (all libraries)                                        47

2514 GETS were required to define 47 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:DMDRIVER/OBJ=OBJ$:DMDRIVER MSRC$:DMDRIVER/UPDATE=(ENH$:DMDRIVER)+EXECML$/LIB
```

0109 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY